

RUI OGAWA

**ESTUDO DE UMA FERRAMENTA DE MEDIAÇÃO DO  
ERRO EM DISCIPLINAS DE ALGORITMOS: UMA  
PROPOSTA DE METODOLOGIA PARA PROMOÇÃO DA  
INTERAÇÃO**

Dissertação apresentada como requisito parcial à obtenção do grau de Mestre. Programa de Pós-Graduação em Informática, Setor de Ciências Exatas, Universidade Federal do Paraná.

Orientadora: Profa. Dra. Leticia Mara Peres  
Coorientador: Prof. Dr. Alexander Robert Kutzke

CURITIBA

2017

RUI OGAWA

**ESTUDO DE UMA FERRAMENTA DE MEDIAÇÃO DO  
ERRO EM DISCIPLINAS DE ALGORITMOS: UMA  
PROPOSTA DE METODOLOGIA PARA PROMOÇÃO DA  
INTERAÇÃO**

Dissertação apresentada como requisito parcial à obtenção do grau de Mestre. Programa de Pós-Graduação em Informática, Setor de Ciências Exatas, Universidade Federal do Paraná.

Orientadora: Profa. Dra. Leticia Mara Peres  
Coorientador: Prof. Dr. Alexander Robert Kutzke

CURITIBA

2017

---

O34e

Ogawa, Rui

Estudo de uma ferramenta de mediação do erro em disciplinas de algoritmos: uma proposta de metodologia para promoção da interação / Rui Ogawa. – Curitiba, 2017.

143 f. ; il. color. ; 30 cm.

Dissertação - Universidade Federal do Paraná, Setor de Ciências Exatas, Programa de Pós-Graduação em Informática, 2017.

Orientador: Leticia Mara Peres – Co-orientador: Alexander Robert Kutzke,.

Bibliografia: p. 121-126.

1. Algoritmos. 2. Matemática – Estudo e ensino – Análise de erros. 3. Programação (Computadores). 4. FARMA-ALG. I. Universidade Federal do Paraná. II. Peres, Leticia Mara. III. Kutzke, Alexander Robert . IV. Título.

CDD: 511.8

---



MINISTÉRIO DA EDUCAÇÃO  
UNIVERSIDADE FEDERAL DO PARANÁ  
PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO  
Setor CIÊNCIAS EXATAS  
Programa de Pós-Graduação INFORMÁTICA

## TERMO DE APROVAÇÃO

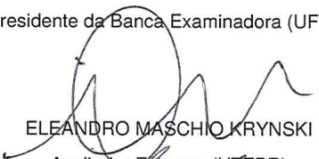
Os membros da Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em INFORMÁTICA da Universidade Federal do Paraná foram convocados para realizar a arguição da Dissertação de Mestrado de **RUI OGAWA** intitulada: **ESTUDO DE UMA FERRAMENTA DE MEDIAÇÃO DO ERRO EM DISCIPLINAS DE ALGORITMOS: UMA PROPOSTA DE METODOLOGIA PARA PROMOÇÃO DA INTERAÇÃO**, após terem inquirido o aluno e realizado a avaliação do trabalho, são de parecer pela sua APROVAÇÃO no rito de defesa.

A outorga do título de mestre está sujeita à homologação pelo colegiado, ao atendimento de todas as indicações e correções solicitadas pela banca e ao pleno atendimento das demandas regimentais do Programa de Pós-Graduação.

Curitiba, 25 de Setembro de 2017.

  
LETICIA MARA PERES

Presidente da Banca Examinadora (UFPR)

  
ELEANDRO MASCHIO KRYNSKI

Avaliador Externo (UFPR)

  
ANDREY RICARDO PIMENTEL

Avaliador Interno (UFPR)



## AGRADECIMENTOS

Em primeiro lugar, a Deus, o criador.

À minha amada esposa, Helena, por todo apoio e compreensão, desde o início desta jornada. Aos meus filhos, Lucas, Victor e Bianca, pois foi pensando no futuro deles, que decidi investir em minha própria formação, não somente por recompensa financeira ou *status*, mas principalmente por acreditar que a educação e a pesquisa são elementos fundamentais e transformadores, que devem ser incentivadas por meio do exemplo. Aos meus pais, Moacir e Tereza, pelo incentivo, desde a tenra idade, a nos interessarmos pelos estudos, pela ciência e pela descoberta. À todos meus familiares e amigos de Sinop. À minha irmã, Cássia e ao meu cunhado, Reginaldo, pelo apoio irrestrito e disponibilidade em cuidar dos nossos filhos durante nossa ausência.

À Universidade do Estado de Mato Grosso – UNEMAT, que proporcionou meu afastamento e todas as condições para que eu me dedicasse aos estudos. Desta instituição, em especial aos professores Rodrigo Bruno Zanin, Giovane Maia do Vale e João Gabriel Ribeiro, pelo incomensurável e irrestrito apoio. Ao amigo Metuzalem Gonçalves Silva, por ter me avisado sobre a oportunidade. Aos professores e amigos Ivan Luiz Pedroso Pires, Elaine Pires e Robson Gomes de Melo, pela calorosa acolhida e preciosos conselhos.

À todos os amigos do curso de mestrado que fiz durante essa caminhada: Josiney de Souza, Maurício Barros, Edmar Bellorini, José Sepka Junior, Ademar Alves, Valber Zacarkim, Jeovane Alves, André Hochuli, Paulo Carvalho, Felipe Bombardelli, Daniel Rezende, Alex Mateus, Renato Mello, Alexandre Alencar, Danilo Faria, Fábio Spanhol, Jefferson Neves, Akihiko Sato, Alexandre Veloso, Alline de Lara, Aramis Fernandes, Cesar Magrin, Cleide Possamai, Daniel Parada, Edemir Maciel, Eduardo Buratti, Jeferson Nichel, Jonivan Oliveira, Josemar Felix, Lauri Laux, Leandro Nagano, Luciano Assumpção, Marina Pimentel, Marlon Lima, Mauricio Requião, Maíra Codo, Patric Forcelini, Pedro Prado, Rafael Canteri, Rita Tesseroli, Rodrigo Ehresmann, Silvio Tacara. Um agradecimento especial ao Diego Pagani, Leandro Pulgatti, Paulo Lisboa, Henrique Shishido, Bruno

Zanette, Carlos Iatskiu e Cristiane Huve, pelas importantíssimas dicas, sugestões e intervenções.

À Coordenação de Pós-Graduação em Informática da UFPR e a todo seu corpo técnico e docente, pela oportunidade e pela busca incessante da excelência. À minha estimada orientadora, Leticia Mara Peres, por todo seu apoio e compreensão, principalmente nos momentos de adversidades. Ao meu coorientador, Alexander Robert Kutzke, cuja tese de doutorado foi principal fonte de inspiração para meu trabalho.

Aos professores do PPGInf, em especial ao Fabiano Silva, Luis Carlos Erpen de Bona, Marcos Alexandre Castilho, Andrey Ricardo Pimentel, Daniel Weingaertner, Elias Procópio Duarte Júnior, Laura Sánchez García, Luis Allan Kunzle e Marcos Didonet Del Fabro.

Principalmente, ao grande Mestre, ao Tutor Inteligente. Jamais conheci um ser humano tão especial como o professor **Alexandre Ibrahim Direne**. Há professores que nos ensinam, outros que nos ajudam, uns que nos orientam, outros que nos cobram. Professor Direne sempre extrapolou esses conceitos. Ele, de modo ao mesmo tempo sereno e subversivo ao sistema vigente, nos inspirou e certamente, continua nos inspirando na busca incessante pela excelência. Não só na pesquisa e na produção científica, mas sobretudo, na vida. Tê-lo conhecido e ter tido a oportunidade e o privilégio de ter sido orientado por ele, mudou completa e profundamente minha visão de ciência, de valores e de vida. Suas palavras sempre muito bem escolhidas, vestidas de todo o rigor científico, mas revestidas de amor, cuidado e, sobretudo, humanismo, nos encorajavam para seguirmos adiante. Professor Direne tinha uma missão, a mais nobre delas, que é educar, mas para além disso, inspirar novas e futuras gerações. Seu legado é eterno. Nos vemos em breve, velho guerreiro.

Muito obrigado!

*A obrigação de produzir aliena a  
paixão de criar.*

---

Raoul Vaneigem

# SUMÁRIO

LISTA DE SIGLAS

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE CÓDIGOS

RESUMO

ABSTRACT

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>Introdução</b>  | <b>15</b> |
| 1.1      | Contexto do projeto . . . . .                                  | 17        |
| 1.2      | Justificativa . . . . .  | 18        |
| 1.3      | Objetivo geral . . . . .                                       | 19        |
| <b>2</b> | <b>Fundamentação teórica</b>                                   | <b>21</b> |
| 2.1      | A Informática como ferramenta de apoio à Educação . . . . .    | 21        |
| 2.2      | Ensino de programação de computadores . . . . .                | 24        |
| 2.3      | Avaliação de ferramentas para mediação de erros . . . . .      | 32        |
| 2.4      | Metodologias de uso para ferramentas interacionistas . . . . . | 34        |
| <b>3</b> | <b>FARMA-ALG - Mediação do erro no ensino de algoritmos</b>    | <b>36</b> |
| 3.1      | Descrição da FARMA-ALG . . . . .                               | 36        |
| 3.2      | Arquitetura da FARMA-ALG . . . . .                             | 37        |
| 3.3      | Análise da FARMA-ALG . . . . .                                 | 42        |
| 3.4      | Considerações para melhoria da ferramenta . . . . .            | 51        |
| 3.5      | Considerações finais sobre a FARMA-ALG . . . . .               | 53        |



|          |   |            |
|----------|---|------------|
| <b>4</b> | <b>Estudos da efetividade da interação</b>                                | <b>54</b>  |
| 4.1      | Objetivos dos estudos da efetividade da interação . . . . .               | 54         |
| 4.2      | Universo da pesquisa . . . . .  | 55         |
| 4.3      | Observações da influência da FARMA-ALG nos resultados de aprendizagem     | 56         |
| 4.4      | Análise crítica das mensagens e comentários . . . . .                     | 65         |
| 4.5      | Análise estatística dos comentários . . . . .                             | 84         |
| 4.6      | Considerações finais sobre os estudos da interação . . . . .              | 97         |
| <b>5</b> | <b>Proposta de metodologia com enfoque na interação</b>                   | <b>99</b>  |
| 5.1      | Pressupostos e estratégias para metodologias em ambientes interacionistas | 100        |
| 5.2      | Atividades do professor na FARMA-ALG . . . . .                            | 105        |
| 5.3      | Fases da metodologia . . . . .  | 107        |
| 5.4      | Considerações finais sobre a metodologia . . . . .                        | 115        |
| <b>6</b> | <b>Considerações finais e trabalhos futuros</b>                           | <b>117</b> |
| 6.1      | Resultados alcançados . . . . .   | 118        |
| 6.2      | Trabalhos futuros . . . . .   | 119        |
|          | <b>BIBLIOGRAFIA</b>   | <b>121</b> |
|          | <b>Apêndice 1 - Lista de questões</b>                                     | <b>127</b> |
|          | <b>Apêndice 2 - Procedimentos no MongoDB</b>                              | <b>135</b> |
|          | <b>Apêndice 3 - Procedimentos no MySQL</b>                                | <b>137</b> |
|          | <b>Apêndice 4 - Procedimentos no R-Studio</b>                             | <b>139</b> |

## LISTA DE SIGLAS

|                  |   |
|------------------|---|
| <b>OA</b>        | Objeto de Aprendizagem  |
| <b>FARMA</b>     | Ferramenta de Autoria para a Remediação de erros com Mobilidade na Aprendizagem               |
| <b>FARMA-ALG</b> | Ferramenta de Autoria para a Remediação de erros com Mobilidade na Aprendizagem de Algoritmos |
| <b>SGBD</b>      | Sistema Gerenciador de Banco de Dados   |
| <b>LEC</b>       | Laboratório de Estudos Cognitivos   |
| <b>MRE</b>       | Múltiplas Representações Externas   |
| <b>STI</b>       | Sistema Tutor Inteligente   |
| <b>AVA</b>       | Ambiente Virtual de Aprendizagem  |
| <b>Moodle</b>    | Modular Object-Oriented Dynamic Learning Environment  |
| <b>ITA</b>       | Intelligent Teaching Assistant  |
| <b>EAM</b>       | Experiências de Aprendizagem Mediadas   |
| <b>CARRIE</b>    | Controlador de Acesso Reflexivo e Retroativo Indexado por Erros                               |
| <b>GC</b>        | Grupo Controle  |
| <b>GE</b>        | Grupo Experimental  |
| <b>IA</b>        | Inteligência Artificial   |
| <b>BOCA</b>      | BOCA Online Contest Administrator   |
| <b>SQL</b>       | Structured Query Language   |

|                |                                  |
|----------------|----------------------------------|
| <b>IHC</b>     | Interface Humano-Computador      |
| <b>WYSIWYG</b> | What You See Is What You Get     |
| <b>CSV</b>     | Common Separated Values          |
| <b>HTML</b>    | HyperText Markup Language        |
| <b>DP</b>      | Desvio Padrão                    |
| <b>CV</b>      | Coeficiente de Variância         |
| <b>ZDP</b>     | Zona de Desenvolvimento Proximal |

## LISTA DE FIGURAS

|      |  |    |
|------|--|----|
| 2.1  | Sequência das atividades de avaliação da FARMA . . . . .   | 33 |
| 3.1  | Arquitetura da FARMA-ALG . . . . .   | 38 |
| 3.2  | Organização do arcabouço . . . . .   | 39 |
| 3.3  | Grafo de similaridade . . . . .  | 41 |
| 3.4  | Opções para Entrar ou Registrar . . . . .  | 44 |
| 3.5  | Mensagem de boas vindas . . . . .  | 44 |
| 3.6  | Tela principal da FARMA-ALG . . . . .  | 45 |
| 3.7  | Interface responsiva . . . . .   | 46 |
| 3.8  | Criação de um novo OA . . . . .  | 46 |
| 3.9  | OA disponível . . . . .  | 47 |
| 3.10 | Introdução de um OA . . . . .  | 47 |
| 3.11 | Questão sobre raízes de uma equação de segundo grau . . . . .  | 48 |
| 3.12 | Casos de teste para raízes de uma equação de segundo grau . . . . .  | 49 |
| 3.13 | Caso de teste para duas raízes . . . . .   | 49 |
| 3.14 | Exemplo de busca no grafo de manipulação . . . . .   | 50 |
| 3.15 | Resultado da busca no grafo de manipulação . . . . .   | 51 |
| 4.1  | Diagramas no DBSchema . . . . .  | 58 |
| 4.2  | Fluxo de respostas e comentários na FARMA-ALG . . . . .  | 59 |
| 4.3  | Fluxograma de análise estatística inferencial . . . . .  | 86 |
| 4.4  | Comparativo entre as tentativas executadas até a primeira interação e desde<br>a interação até o acerto . . . . .              | 91 |
| 4.5  | Boxplot e Plotmeans comparando as tentativas executadas até a primeira<br>interação e desde a interação até o acerto . . . . . | 92 |
| 4.6  | Comparativo entre o tempo até a primeira interação e desde a interação<br>até o acerto . . . . .                               | 94 |

|     |  |     |
|-----|--|-----|
| 4.7 | Boxplot e Plotmeans comparando os tempos até a primeira interação e desde a interação até o acerto . . . . . | 95  |
| 5.1 | Zona de Desenvolvimento Proximal . . . . .   | 102 |
| 5.2 | Estágios da Zona de Desenvolvimento Proximal . . . . .   | 103 |
| 5.3 | Fases da metodologia . . . . .   | 107 |
| 5.4 | Filtros e resultado de similaridade de questões . . . . .  | 113 |
| 5.5 | Resultado da busca por tags de erros . . . . .   | 114 |

## LISTA DE TABELAS

|     |  |     |
|-----|--|-----|
| 4.1 | Turmas analisadas . . . . .  | 55  |
| 4.2 | Tabelas importadas para o MySQL . . . . .                                | 61  |
| 4.3 | Questões ordenadas por Listas de Exercícios . . . . .                    | 62  |
| 4.4 | Respostas por questões . . . . .   | 63  |
| 4.5 | Respostas por listas de exercícios . . . . .                             | 63  |
| 4.6 | Tempo e tentativas após intervenções . . . . .                           | 83  |
| 4.7 | Média, Desvio Padrão e Coeficiente de Variância das tentativas . . . . . | 91  |
| 4.8 | Média, Desvio Padrão e Coeficiente de Variância do tempo . . . . .       | 94  |
| 5.1 | Objetivos e produtos das fases . . . . .                                 | 108 |
| 6.1 | Lista das questões com enunciados . . . . .                              | 127 |

## LISTA DE CÓDIGOS

|     |  |     |
|-----|--|-----|
| 4.1 | Conversão de temperaturas (tentativa 1 - erro de saída) . . . . .  | 71  |
| 4.2 | Conversão de temperaturas (tentativa 41 - erro de saída) . . . . . | 72  |
| 4.3 | Código que apresenta respostas aleatórias . . . . .                | 76  |
| 6.1 | Script de exportação de MongoDB para CSV . . . . .                 | 135 |
| 6.2 | Consultas no MySQL . . . . .                                       | 137 |
| 6.3 | Rotina do R-Studio . . . . .                                       | 139 |

## RESUMO

O objetivo deste trabalho é propor uma metodologia direcionada ao aumento da interação entre professores e alunos que usam a FARMA-ALG. Para alcançar esse objetivo, foi realizada uma pesquisa usando a abordagem sociointeracionista do relacionamento professor aluno através de um ambiente virtual de mediação do erro escolar no ensino de algoritmos e a possível melhoria do seu processo pedagógico por meio de estudos empíricos, análises estatísticas descritivas e inferenciais. Na fundamentação teórica são abordados os aspectos sobre a importância do ensino de programação, em especial da disciplina de algoritmos. São feitas considerações sobre a desistência e a evasão de alunos dos cursos de Computação, sua associação com o erro escolar e com as relações das intervenções sociointeracionistas dos professores com relação às respostas submetidas por alunos, realizadas por meio de comentários. É mostrada a importância do erro escolar e sua mediação com uso de sistemas computacionais, bem como da interação do professor com o aluno. Para obtenção de informações sobre a experiência de uso da ferramenta FARMA-ALG pelo professor, foi realizada uma pesquisa empírica, cujos métodos são descritos nos fundamentos da solução adotada. Também são explicados os métodos e técnicas de análise estatística descritiva e inferencial que possibilitam buscar padrões indicativos da correlação da interação entre professores, alunos e os níveis de acerto. São apresentados os resultados, a relação entre as informações obtidas e o processo de interação. Ao final, é proposta uma metodologia para utilização da ferramenta pelo professor, com enfoque em suas características interacionistas e apresentados os resultados e novas ideias de pesquisa futura.

**Palavras-chave:** mediação, sociointeracionismo, erro escolar, ensino-aprendizagem, algoritmos, metodologia, FARMA-ALG.



## ABSTRACT

This dissertation has its focus on proposing a methodology aimed to increase the interaction amid lectures and students using FARMA-ALG. A research of this interaction was conducted using a social-interactionist approach through a virtual environment of learning error on the algorithmic subject to achieve that goal. It was also researched possible improvements of pedagogical process, by means of empirical studies, descriptive and inferential statistical analysis. On the theoretic fundamentaion, are addressed aspects of the importance of programming teaching, in particular algorithms. Considerations are made on withdrawals and dropouts of the computational lecture, its association with learning error and its relation amid social-interactionists interventions from lecturers with answers submitted by students, made through comments. The importance of learning error and its mediation is shown with the use of computational systems, as well as the lecturer-student interaction. Information on the user experience of FARMA-ALG by the lecturer was gathered through an empirical research, whose methods are described on the fundamentals of the adopted solution. Also explained are methods and techniques of descriptive and inferential statistical analysis which allows search of indicative patterns of interaction between lecturers, students and rate of success. Results are presented, as well as the relation between gathered information and interaction process. At the end, a methodology for the lecturer's use of the tool is proposed, focusing on its interactionist characteristics. Results are presented and new research ideas are proposed.

**Keywords:** mediation, social-interactionism, learning error, teaching-learning, algorithms, methodology, FARMA-ALG.

## CAPÍTULO 1

### INTRODUÇÃO

Ferramentas de autoria são softwares que permitem ao tutor humano manipular (criar, alterar ou excluir) sessões de ensino, de modo a compor um software para ensino e aprendizagem de algum conteúdo (MARCZAL et al., 2015). Há diversas formas sob as quais um conteúdo pode ser disponibilizado aos alunos. Considerando uma ferramenta de autoria, a materialização dos conteúdos pode se dar sob a forma de Objetos de Aprendizagem (OAs), que são concebidos e construídos pelo professor. Objetos de aprendizagem são definidos aqui como qualquer entidade, digital ou não digital, que pode ser usada, reutilizada ou referenciada durante a aprendizagem com suporte tecnológico (WILEY, 2003).

Há diferentes tipos de OAs para dar suporte à aprendizagem, que podem ser interativos ou não-interativos. OAs não-interativos consistem, por exemplo, no esboço de uma lição ou correspondente apresentação, na forma de uma sequência de slides organizados em estrutura de tópicos, mostrando conceitos e exemplos relevantes (KOOHANG; HARMAN, 2007, p. 565). Segundo Moreno e Mayer (2007 apud FLÔRES; TAROUCO, 2008), há vários tipos de ambientes interativos. Existem ambientes que fornecem linhas de orientação, aconselhamento, *feedback* e apoio adequado às necessidades do aluno sob a forma de exercícios e questões. A forma como será a interação do aluno com os exercícios e as questões propostas, dependerá de como estes OAs foram criados e como é realizado o tratamento de erros pela ferramenta de autoria. Os agentes pedagógicos podem servir como tutores que, além de ajudar os alunos a aprender, os auxiliam a encontrar estratégias para uma aprendizagem eficaz.

Dentre as estratégias, encontra-se a remediação, que é, essencialmente, a apresentação de uma mensagem de auxílio ao aprendiz (RAMOS, 2011). Todavia, de acordo Leite et al. (2015), na maioria dos OAs, a remediação de erros tem sido aplicada de forma mais restrita. Normalmente, disponibilizam-se os OAs aos alunos pelo sistema com pouca ou nenhuma

consideração sobre a possibilidade de retroação ao estado de erro, nem sua mediação. As tentativas de mediação, nesses casos, concentram-se apenas em proporcionar um retorno simples, por meio de mensagens, se a interação está correta ou incorreta (MARCZAL; DIRENE, 2011).

É importante, neste momento, explicitar a diferença entre mediação e remediação. Quando se fala de mediação, a referência é sobre uma abordagem intervencionista, que defende que é por meio da ação e das interações do sujeito com o meio social e a cultura que ocorre o desenvolvimento das pessoas e da própria atividade (CASSANDRE; QUEROL, 2014). Sob a luz dos estudos de Vygotsky (1997), Martins (2014) aponta que a mediação é uma interposição que provoca mudanças, que promove o desenvolvimento através de uma intencionalidade socialmente produzida. A remediação, por sua vez, consiste na imediata correção do erro, fornecendo rápido retorno sobre os erros cometidos pelo aluno, para que ele possa ser orientado ao longo do processo e não somente na etapa final (ANDERSON, 2013).

Diante da necessidade de ferramentas de autoria que ofereçam suporte à mediação de erros no ensino de Matemática, Marczal (2014) propôs a Ferramenta de Autoria para a Remediação de erros com Mobilidade na Aprendizagem (FARMA)<sup>1</sup>. Esta ferramenta é um arcabouço computacional com abordagem sociointeracionista para tratamento do erro escolar. Nela, a qualquer momento, tanto o professor quanto o aprendiz (ou ambos, juntos) podem restaurar a sessão exata do momento em que o erro ocorreu e, com isso, explorá-lo em mais detalhes. Esse mesmo mecanismo de retroação permite também que o exercício seja refeito, possibilitando que o aprendiz reveja seu erro e também possa tentar refazer o exercício (MARCZAL, 2014). Isso é possível devido ao componente de armazenamento e recuperação de submissões das respostas dos alunos, constituindo uma base que pode ser acessada, manipulada e pesquisada.

O erro escolar deve ser entendido sob uma perspectiva mais ampla, que considere a unidade de suas relações e inter-conexões com outros erros e dados. Os registros de respostas se relacionam entre si de diferentes maneiras; isto é, respostas são similares à

---

<sup>1</sup> <<http://farma.educacional.mat.br/>>

outras sob diferentes aspectos (por exemplo, sintáticos ou semânticos), trabalhando sobre as relações de similaridade existentes entre diferentes respostas. A FARMA-ALG<sup>2</sup> propõe, através da análise de tais relações, que seja possível captar propriedades relevantes para a mediação do erro (KUTZKE; DIRENE, 2014).

Esta pesquisa tem como proposta central estudar como professores e alunos interagem por meio da FARMA-ALG, com o objetivo de propor mecanismos para promover o aumento dessa interação. O trabalho é conduzido com base em uma pesquisa empírica com método qualitativo e técnicas de estatística descritiva e inferencial, sobre como o professor interage com as turmas utilizando os mecanismos de comentários oferecidos pela ferramenta. O escopo do presente trabalho se restringe à ferramenta FARMA-ALG, cujo funcionamento é explicado no Capítulo 3, concentrando-se no estudo do aumento da interação entre professor e aluno.

## 1.1 Contexto do projeto

Marczal (2014) desenvolveu uma ferramenta de autoria web que possibilita que o erro seja utilizado como oportunidade de aprendizado através de mecanismos de remediação e retroação. A pesquisa de Kutzke e Direne (2015) visa a instrumentalização de professores e alunos na busca da mediação do erro durante o trabalho educativo, integrando o erro à dinâmica de formação e desenvolvimento de conceitos científicos dos alunos.

Em ambas as pesquisas há o enfoque nos aspectos do erro escolar, sua mediação e como ela pode ser útil no processo de aprendizado. Verifica-se também, nos trabalhos dos autores, forte preocupação com a abordagem sociointeracionista, que, segundo Soares (2005), concebe a aprendizagem como um fenômeno que se realiza na interação com o outro. A interação possibilitada pelo sistema incentiva o aprendizado colaborativo através da mediação e interação entre professores e alunos.

Marczal (2014, p. 36) destaca que *“a aprendizagem se torna resultado da interação social a partir do compartilhamento de significados com o objetivo de suprir, por meio da colaboração, o conhecimento faltante, tanto para o professor quanto para o aluno”*. Em

---

<sup>2</sup> <<http://farma-alg.com.br>>

consonância com a importância da interação, Kutzke e Direne (2015) salientam que *“sem a mediação do erro, professores não possuem condições ideais para refletir sobre seu próprio método de ensino, e os alunos, por sua vez, ficam sem acesso aos seus erros e da mediação dos mesmos pelo professor”*.

É clara a atenção dos autores dedicada à mediação do erro e à interação entre docentes e seus aprendizes. De fato, os resultados das pesquisas de ambos mostram que houve melhoria no processo de aprendizagem dos alunos através da utilização dos sistemas propostos. Entretanto, embora os resultados sejam positivos e animadores considerando o ensino-aprendizagem, não é possível afirmar em quanto sua utilização contribuiu no aumento da interação entre os alunos e seus professores. Para além do estudo sobre o aumento da interação, são desconhecidas também a influência e as consequências do aumento dessa interação. Também é desconhecida qualquer forma de documentação ou metodologia que seja capaz de guiar de modo efetivo os professores e alunos durante a utilização da ferramenta.

## 1.2 Justificativa

Percebe-se, nesse sentido, a necessidade de estudar o aumento da interação que tais ferramentas proporcionaram entre seus utilizadores. Além disso, há uma carência de metodologias que guiem a criação de novas formas de projeto instrucional sob um modelo sociointeracionista, baseado na interação por meio da tecnologia. Sem um guia que oriente o docente na correta e efetiva utilização de ferramentas computacionais com foco no interacionismo, os resultados de seus esforços podem ficar aquém do que poderiam ser.

Considerando uma ferramenta computacional como a FARMA-ALG, que oferece a retroação ao contexto de erro e permite sua mediação através da interação entre alunos e professores, cabem aqui algumas perguntas:

- Como ela pode ajudar o professor em sua prática pedagógica na disciplina de algoritmos?
- O quanto ela pode auxiliá-los para aumentar a interação com seus alunos na disciplina

de algoritmos?

- O modo como professores e alunos utilizam os recursos de mensagens e comentários da ferramenta pode fornecer informações que venham a contribuir com o aumento da interação?
- O aumento da interação entre professores e alunos efetivamente contribui para a melhoria do processo de ensino-aprendizagem de algoritmos?
- O professor está preparado para utilizar a ferramenta a contento, considerando seus pressupostos de educação intervencionista?

Caso a interação do docente possa ser capaz de reduzir a quantidade de tentativas, a efetividade da FARMA-ALG em seu propósito de intervenção sociointeracionista estaria então comprovada. Ao final foi proposta uma metodologia de utilização da FARMA-ALG que contemple os pressupostos sociointeracionistas para que os professores possam ter um guia sobre como utilizar a plataforma da maneira mais eficaz possível. Essa metodologia visa garantir a efetividade na utilização da FARMA-ALG pelos professores, para que mesmo sem o auxílio de outro professor que já tenha domínio da ferramenta, o uso seja orientado do modo correto.

### 1.3 Objetivo geral

O objetivo deste trabalho é propor uma metodologia direcionada ao aumento da interação entre os professores e alunos que usam a FARMA-ALG. Esta metodologia deve ser genérica e replicável para que a utilização da plataforma pelos professores e seus alunos atinja seu objetivo interacionista com sucesso. Através de estudos empíricos e posteriormente, estatísticos, foi pesquisado como é a interação do professor com seus alunos ao utilizar a ferramenta e como isso impacta sobre a quantidade e qualidade das respostas dos alunos às questões.

### 1.3.1 Objetivos específicos

- Estudar a arquitetura da FARMA-ALG;
- Estudar a interação entre alunos e professores nos registros históricos da FARMA-ALG, por meio de análise estatística descritiva e inferencial das mensagens de interações entre professores e alunos;
- Propor uma metodologia de utilização da FARMA-ALG pelos professores, para que os objetivos interacionistas sejam alcançados, melhorando os resultados e o processo de ensino-aprendizagem por meio do aumento quantitativo e qualitativo de interações.

## **CAPÍTULO 2**

### **FUNDAMENTAÇÃO TEÓRICA**

Este capítulo versa sobre as questões inerentes à fundamentação teórica do presente trabalho. A Seção 2.1 aborda os primeiros registros do uso da Informática como ferramenta de apoio à Educação, conceitos sobre ambientes de autoria e objetos de aprendizagem e considerações sobre o erro escolar. Na Seção 2.2.1 são abordados os aspectos sobre interação, desenvolvimento cognitivo e sistemas com suporte ao erro escolar, destacando a necessidade de sistemas de retroação ao estado de erro. A Seção 2.2 apresenta uma introdução sobre aspectos do ensino de programação de computadores, abordando aspectos relacionados à disciplina de algoritmos. Também são revisados os desafios sobre desistência e evasão escolar de alunos e a importância do erro como parte do processo cognitivo, além das ferramentas computacionais disponíveis que possuem uma abordagem direcionada ao erro. Ao final deste capítulo, na Seção 2.3, é discutida a importância em avaliar a efetividade de ferramentas para mediação de erros.

#### **2.1 A Informática como ferramenta de apoio à Educação**

No Brasil, os primeiros registros sobre o uso da Informática aplicada à Educação, sob uma perspectiva construtivista e construcionista, datam do final de década de 1970. Experimentos realizados pelo Laboratório de Estudos Cognitivos (LEC), da UFRGS exploravam a potencialidade do uso do computador com a linguagem Logo (NASCIMENTO, 2007, p. 13). Segundo o autor, esses trabalhos procuravam compreender o raciocínio lógico-matemático das crianças, possibilitando a intervenção como forma de promover a aprendizagem autônoma. Esses experimentos foram apoiados nas teorias de Jean Piaget e Seymour Papert, sobre cujos conceitos o presente trabalho tem forte embasamento.

Ferreira e Duarte (2012) identificam uma forte hegemonia nos trabalhos relacionados à Informática na Educação sobre a importância das pedagogias do aprender a aprender. Os



autores tecem críticas à dicotomia presente nas discussões sobre a incompatibilidade entre o processo de ensino – o qual prioriza a transmissão do conhecimento – e da aprendizagem, que preconiza exatamente o aprender a aprender como parte essencial do processo de formação do conhecimento. Embora Ferreira e Duarte sustentem que a transmissão de conhecimentos do professor para o aluno seja elemento indispensável para construção de novos conhecimentos pelos alunos, é importante destacar que, sobretudo na área de programação de computadores, *“é mais importante que o aluno consiga desenvolver um método de aquisição, elaboração, descoberta e construção de conhecimentos, do que aprender os conhecimentos que foram descobertos e elaborados por outras pessoas”* (DUARTE, 2001, p. 35).

Essa capacidade do ser humano de monitorar e auto-regular seus processos cognitivos é nomeada, na área da Psicologia Cognitiva de *metacognição* (NOUSHAD, 2008). Quando se aprofunda na definição de metacognição, é verificado que esta se refere não somente ao conhecimento que o indivíduo tem sobre seu próprio conhecimento, mas também sobre seus próprios processos cognitivos (FIGUEIRA, 2003). O aprender a aprender é portanto, a cognição da cognição, de modo que metacognição é um novo termo para uma ideia antiga (BROWN, 1987).

### **2.1.1 Múltiplas representações externas (MREs) para confecção de OAs inteligentes em ambientes de autoria**

Marczal e Direne (2012) consideram os ambientes de autoria como uma das formas mais conhecidas e difundidas para a aplicação da Informática na educação, pois permitem organizar ou produzir uma sequência de informações de modo a compor um software para o ensino-aprendizagem que possibilite a interação do usuário com um domínio específico. Alguns ambientes de autoria têm como objetivo a criação de Sistemas Tutores Inteligentes (STIs). Marczal et al. (2015) citam as ferramentas RUI (DIRENE, 1997), REDDEEM (AINSWORTH; FLEMING, 2006) EON (MURRAY; WOOLF; MARSHALL, 2004) e SIMQUEST (SILVA; GERMANO; MARIANO, 2011) como exemplos importantes, embora exijam do autor certo conhecimento de programação de computadores, o que demanda

tempo e esforço dos professores.

Há ambientes que apresentam características interessantes, como a construção de conteúdos que podem ser acessados por meio de plataformas colaborativas na web. É o caso do EDUCA, que permite que o material possa ser criado pelo professor e atualizado pelos próprios alunos (CABADA; ESTRADA; GARCÍA, 2011). O software CTAT, de Alevén e Sewall (2010) traz uma abordagem inovadora através da utilização de grafos de referência para guiar o aluno pelo problema. O CTAT também oferece mecanismos de remediação do erro, através do qual o autor pode intervir com mensagens sobre esses erros, por exemplo.

Os resultados dos trabalhos em Ambientes de Autoria costumam culminar em elementos pedagógicos que possam ser utilizados pelos alunos. Esses elementos são denominados de Objetos de Aprendizagem (OAs). Há várias definições para o conceito OAs que normalmente estão atreladas aos aspectos pedagógicos ou técnicos do domínio da disciplina que está sendo abordada. Segundo Marczał (2014), o conceito de OAs envolve a fragmentação de um determinado currículo em partes muito bem especificadas e definidas. Essas partes recebem metadados e são armazenadas em repositórios digitais de forma que, através dos metadados, possam ser facilmente encontradas e reutilizadas.

Considerando que seus conteúdos são pequenos e restritos, uma característica peculiar que os OAs podem oferecer consiste no fato de que se torna possível avaliar como o aluno está aprendendo. Para tanto, é necessário que se pense em OAs inteligentes, que se comportem de modo exclusivo para cada aprendiz, de acordo com seu nível de interação. Dentre as diversas características de um OA, Marczał destaca a possibilidade, ainda pouco explorada, de permitir que os OAs forneçam subsídios somativos à formação do aluno. Em outras palavras, é possível verificar se os alunos estão conseguindo atingir os objetivos propostos e, caso necessário, oferecer auxílio por meio de intervenções sobre sua interação com o conteúdo. Para que se alcance esses objetivos, surge uma nova geração de Sistemas Tutores Inteligentes, que propicia a utilização de OAs que contemplem os aspectos relacionados ao erro escolar.

Retomando a abordagem metacognitiva, discutida anteriormente, o conceito de MREs

de Ainsworth (2006) fornece as características necessárias para um aprendizado intuitivo. Para Ainsworth, há cinco fatores que devem ser considerados em um ambiente que utilize MREs:

1. o número de representações;
2. a maneira como a informação está distribuída;
3. a forma das representações;
4. a sequência das representações e
5. o suporte para a tradução entre as representações.

As tarefas cognitivas envolvidas na manipulação de representações externas devem:

1. entender a forma da representação;
2. criar intuitivamente a relação entre a representação e o domínio representado;
3. escolher uma representação apropriada entre várias disponíveis e
4. construir uma nova representação apropriada se for requisitada.

Compreender o modelo de MRE não é tarefa trivial e consiste em uma sequência que se inicia com a representação verbal, passando pela representação visual, com o objetivo de estimular a atuação por intermédio da interação reflexiva do aprendiz sobre o conteúdo específico (DIRENE et al., 2009). Dá-se ênfase no controle do aprendiz sobre o conteúdo, de forma que os aspectos metacognitivos se tornem mais poderosos, o que permite uma intervenção contextualizada. Para Maschio (2013), isso delega mais responsabilidades a quem aprende sobre o próprio aprendizado, permitindo que o desempenho sobre o que é ensinado seja influenciado positivamente.

## **2.2 Ensino de programação de computadores**

O principal objetivo do ensino de programação de computadores é capacitar os alunos para que eles possam conceber bons programas. Um programa de computador consiste na concretização de uma solução, a partir da implementação de um algoritmo em uma determinada linguagem de programação, para um determinado problema. Esse algoritmo

é uma sequência lógica de comandos, que posteriormente podem ser executados ou interpretados em um computador. Planejar e organizar essa estrutura lógica é essencial para a obtenção de resultados satisfatórios, pois a não definição prévia do que cada linha do programa fará e em que sequência será executada, certamente resultará em um programa ineficiente, de difícil compreensão e manutenção.

O aprendizado de programação exige um alto nível de abstração e poder cognitivo do aluno para que ele compreenda as estruturas do mundo real que deverão ser representadas em código (KUTZKE; DIRENE, 2015). Aprender a programar com êxito exige do aluno o que é denominado pensamento abstrato ou lógico-discursivo, necessário para o aprendizado de algoritmos, pois se apoia em conceitos e raciocínios abstratos, operando fundamentalmente por mediação (MARTINS, 2014). O desempenho do aluno nas disciplinas de algoritmos está, portanto, diretamente relacionado com o seu domínio desse tipo de pensamento. Para ensinar os princípios básicos necessários às disciplinas de programação, os conceitos de algoritmos normalmente são ministrados no início dos cursos de graduação. Tal prática se justifica tendo em vista que os princípios essenciais de lógica são a base para o ensino destas disciplinas, pois desenvolvem as habilidades de análise e resolução de problemas (RAPKIEWICZ et al., 2006, p 1).

Nos cursos de Computação e engenharias, as disciplinas de algoritmos e estrutura de dados são de extrema importância para que o aluno tenha uma base sólida, de forma que assegure seu bom desempenho no decorrer do currículo. Entretanto, geralmente o tempo reservado para essas disciplinas é restrito, o que compromete o bom aproveitamento do aluno nas disciplinas subsequentes. De acordo com Maschio (2013) até mesmo o desenvolvimento da perícia do futuro egresso pode ser influenciada, pois todo o legado do trabalho nas disciplinas introdutórias de computação pode afetar sua prática profissional. Embora o egresso possa buscar aprimoramento complementar após sua graduação, trabalhar a capacidade cognitiva dos alunos é determinante para o desenvolvimento das habilidades necessárias para as disciplinas introdutórias com o objetivo de proporcionar as melhores formação e perícia possíveis. Segundo Krynski (2007), os sistemas educacionais na área da computação tendem a tratar o ensino de programação por meio de experimentos com

linguagens formais e destaca que ainda há certa dificuldade do aluno em trabalhar com essas linguagens, o que pode gerar dificuldades e desmotivação, ocasionando até mesmo desistências e evasões. Para que seja possível conceber STIs que contemplem os aspectos do erro escolar, primeiramente é fundamental compreender a origem do erro escolar na área da Computação.

### 2.2.1 Cognição, interação e STIs com ênfase no erro escolar

A atividade de aprender a programar exige elevado nível de cognição e abstração por parte do aprendiz. Os estudos de Vygotsky (1989), indicam que o estímulo do desenvolvimento cognitivo do aluno para obter um conhecimento sólido e aprofundado está intimamente relacionado com a interação entre os indivíduos. Com base nas afirmações de Vygotsky, para que o aluno possa ter desenvolvimento cognitivo, é desejável que haja mediação suficiente para promover a maior interação possível entre os envolvidos no processo de ensino e aprendizagem – os professores e seus alunos.

Aumentar a interação entre professores e alunos é uma das estratégias propostas por Chickering e Ehrmann (1996) quando se trata da utilização de recursos computacionais na Educação. Diante de uma situação de erro do aluno, a interação do professor, através de comentários, pode contribuir com a elucidação de problemas comuns e assim, facilitar sua compreensão e resolução. Alinhado com uma abordagem interacionista, Marczał (2014) propõe, na ferramenta FARMA, a possibilidade de analisar a interação de cada aprendiz com a ferramenta e o professor, principalmente no que concerne aos erros cometidos por eles. Essa interação é apoiada sobre a teoria sociointeracionista de Vygotsky (1980), permitindo que alunos e professores troquem mensagens sobre erros cometidos. Para o sociointeracionismo, *“o desenvolvimento se produz não apenas por meio da soma de experiências, mas, e sobretudo, nas vivências das diferenças, pois o aluno aprende imitando, concordando, fazendo oposição, estabelecendo analogias, internalizando símbolos e significados, tudo isto num ambiente social e historicamente localizado”* (MARTINS, 1997, p. 120).

Alguns STIs voltados para ensino de programação para novatos passaram a oferecer

recursos para que o erro cometido pelo aluno pudesse ser detectado e diagnosticado (ADAM; LAURENT, 1980; LUKEY, 1980; HASEMER, 1983; JOHNSON; SOLOWAY, 1985; SHAPIRO, 1982). Os métodos de diagnóstico dos erros em programação foram classificados em três categorias por Boulay e Sothcott (1988), descritas a seguir: soluções por referências, por análise de especificação e por diálogo de depuração.

Na categoria de diagnóstico por solução de referência, em que a solução proposta pelo aluno é confrontada com a do professor, Adam e Laurent (1980) citam o sistema LAURA. Três sistemas foram mencionados na categoria de diagnóstico por análise de especificação: Lukey (1980) discorre sobre o sistema PUDSY. Hasemer (1983) fala acerca do sistema AURAC e Johnson e Soloway (1985) abordam o sistema PROUST. O funcionamento deles é baseado na descrição e especificação de metas para as repostas. A terceira categoria utiliza o diálogo de depuração, que dialoga passo a passo com o estudante durante a resolução do problema. Shapiro (1982) estudaram o sistema SHAPIRO, pertencente à essa última categoria, que efetua diagnósticos de códigos em Prolog.

Entretanto, ainda que os sistemas citados realizem por consequência o diagnóstico dos erros, não há um mecanismo que permita o tratamento desses erros. Considerando a adoção de STIs para o ensino de algoritmos, é interessante pensar na concepção de plataformas que aproveitem o erro como recurso para a aprendizagem, pois pesquisas comprovaram a eficácia de sistemas nos quais a utilização de uma abordagem de análise e resolução de problemas com exemplos errôneos resultaram em ganho do desempenho escolar dos alunos (ISOTANI et al., 2011) e (MCLAREN et al., 2012). Como exemplos de STIs que contemplem a questão do erro, o conceito de MREs de Ainsworth (2006), foi utilizado por Leite, Pimentel e Oliveira (2011) para remediação de erros em Objetos de Aprendizagem (OAs). Posteriormente, os mesmos autores apresentam um estudo sobre remediação de erros em Matemática. Avançando no campo da retroação ao contexto de erro, Marczal et al. (2015) propuseram a FARMA, para retroação a contextos de erros em Matemática. Kutzke e Direne (2014), por sua vez, propuseram a FARMA-ALG, para retroação e remediação de erros em exercícios de algoritmos.

É necessário revisar algumas considerações sobre o erro escolar, uma vez que o erro deve

ser visto como parte do processo de aprendizado. Segundo SFORNI (2004), é necessário ver o erro como um momento em que fica evidente uma situação de conflito que pode levar à aprendizagem de um novo conceito. Uma das consequências negativas é que o erro pode gerar um sentimento de incapacidade no aluno, levando-o à desistência das disciplinas de algoritmos e consequente possível evasão dos cursos de Computação. O professor, uma vez instrumentalizado com o apoio de uma ferramenta computacional de apoio à contextos de erro, pode ter a oportunidade de contribuir na redução dos casos de desistências, trancamentos e evasões.

### **2.2.2 Desistência e evasão nos cursos de Computação**

A crescente demanda por profissionais de Computação tem gerado grande procura pelos cursos da área, entretanto, a evasão deles não é um fato recente e tem sido estudada por vários autores (GIRAFFA; MORA, 2013). O pensamento abstrato e o raciocínio lógico, imprescindíveis para a aprendizagem de algoritmos, são habilidades que eventualmente não puderam ser trabalhadas de forma efetiva com todos os alunos durante sua formação antes do ingresso no ensino superior. Tal fato tem sido apontado como causador de desistências e cancelamentos nas disciplinas de algoritmos e consequente evasão dos cursos de Computação (GOMES; HENRIQUES; MENDES, 2008).

A disciplina de algoritmos exige sólidos conhecimentos de raciocínio lógico, necessários para que o aluno não incorra em erros considerados primários durante a resolução de exercícios. Quando o aluno passa por dificuldades nesse aspecto, isso pode contribuir para seu baixo desempenho nas avaliações, fazendo-o sentir-se desestimulado e propenso a desistir. Trata-se de um problema complexo, que envolve aspectos sobre o fracasso escolar na educação básica dos alunos, com causas que podem advir de uma ou mais situações. Variações dos procedimentos de avaliação em períodos diferentes, rotatividade de docentes, qualidade do corpo docente, políticas de atribuições de aulas, relações hierárquicas entre professores, relações entre educadores e autoridades governamentais em determinada conjuntura política, nível de satisfação dos professores com suas condições de trabalho e do alvo que eles elegem para sua insatisfação, são alguns dos motivos que Patto (1988,

p. 77) destaca em sua pesquisa.

O professor da disciplina de algoritmos pode ajudar esses alunos identificando suas principais dificuldades e oferecendo apoio pedagógico. Entretanto, essa tarefa requer muito tempo e esforço do professor, principalmente nas turmas iniciais que costumam ser numerosas. O advento de sistemas computacionais que promovam a mediação do erro cometido pelo aluno pode contribuir para que o professor ofereça auxílio de forma mais abrangente e ao mesmo tempo, personalizada.

### 2.2.3 Sistemas com suporte a mediação do erro escolar

A importância da mediação do erro tem ganhado relevância e vem se tornando um recurso cada vez mais explorado pelos Ambientes Virtuais de Aprendizagem (AVAs). Embora não sejam tão específicos quanto os mecanismos de mediação propostos pela FARMA e FARMA-ALG para turmas de matemática e de algoritmos respectivamente, é preciso lembrar que os AVAs considerados de uso geral, (por exemplo, Moodle<sup>1</sup>), possibilitam um certo nível de mediação. Os fóruns de discussão desses AVAs, por exemplo, permitem que o professor-tutor se aproxime dos cursistas para mediar e inclusive, apontar erros. É importante enfatizar que, embora esta seção aborde as características das ferramentas de mediação do erro para disciplinas específicas, particularmente a matemática e a programação de computadores, essas não são as únicas soluções que ofertam mediação.

Estanislau (2014) realizou uma análise pedagógica da mediação docente no fórum de discussão de um curso no AVA Moodle. A autora nota que a mediação se faz efetiva no fórum através de três principais intervenções mediadoras:

- as indicações e complementações bibliográficas, que potencializam o estudo;
- o encaminhamento das discussões a partir de experiências próprias e reorientação com relação às falhas ocorridas durante o fórum, como forma de acolhimento; e
- a referência aos sujeitos do fórum como constituintes de uma rede de aprendizagem, estimulando a dinâmica colaborativa.

A docente, ao utilizar o recurso de fórum do Moodle, amparou-se em uma pedagogia

---

<sup>1</sup> <<https://moodle.org>>



mediadora, visando a autonomia do aprendiz, a problematização, a dialogicidade, a construção coletiva do conhecimento e a motivação. A autora afirma que os fóruns de discussão são importantes ferramentas de práticas pedagógicas potencializadoras da mediação docente online. Além do papel do professor como mediador nos fóruns de discussão, pesquisado por Estanislau, é importante mencionar que estudos comprovam a importância dessa mediação pela perspectiva dos alunos.

Nesse sentido, os estudos de Machado e Teruya (2009, p. 1733) nos mostram que a mediação em um AVA só é efetiva com forte interação do tutor do curso, pois é o tutor quem orienta o aluno nesses novos espaços virtuais, dando significado aos conteúdos. As autoras identificaram nas respostas dos alunos uma cobrança sobre a postura que o tutor deveria ter. Aspectos como a afetividade, quando o tutor cita o aluno pelo nome, comentários às respostas dos alunos e sensibilidade do tutor para perceber quando o aluno está “distante” do curso, foram citados pelos entrevistados. A mediação pedagógica por meio dos fóruns de discussões em AVAs é possível, desde que o tutor possua conhecimento aprofundado sobre o assunto, tenha clareza nas explicações, demonstre segurança e empatia nas discussões e principalmente, interaja para que o aluno não se sinta excluído (MACHADO; TERUYA, 2009).

Entre os sistemas computacionais de apoio ao ensino de algoritmos que oferecem recursos automáticos de mediação do erro, há os trabalhos de Leite, Pimentel e Pietruchinski (2012) e Leite, Pimentel e Oliveira (2011), que fazem uso da abordagem de MREs (AINSWORTH, 2006). Posteriormente o trabalho de Leite e Pimentel (2013) trata da definição de uma arquitetura específica para remediação de erros matemáticos aplicados a OAs, com o uso de STIs.

Foram encontradas duas ferramentas para classificação de erros, que favorecem e direcionam sua mediação, constituindo importante recurso para a informática educativa. Um modelo para classificação de erros de matemática é apresentado no trabalho de Leite, Pimentel e Oliveira (2011). O estudo de Ramos (2011), por sua vez, apresenta um mecanismo para classificação de erros de indução analítica.

Os sistemas que contemplam a retroação a contextos de erros e sua mediação podem

ajudar alunos e professores a usar o erro como forma de identificar partes equivocadas dentro do processo cognitivo do aluno e então corrigi-las. Nesse sentido, Marczal (2014) destaca que uma das atividades principais do professor na FARMA, é a promoção de discussões, possibilitando que o professor troque mensagens com os alunos e assim possam discutir sobre seus erros. Também seguindo a proposição interacionista, a FARMA-ALG possibilita a interação entre professores e alunos através da troca de comentários.

Além dos aspectos sobre a retroação ao contexto de erro, sua mediação e a importância da interação entre professor e aluno, também é importante avaliar como é a experiência de uso do sistema pelo professor. As pesquisas sobre STI têm priorizado seu uso pelos alunos, entretanto, auxiliar professores e instrutores a lecionar melhor é uma atividade tão importante quanto ensinar os alunos (YACEF, 2002). Essa visão, relativamente recente, tem gerado uma nova sub-área dos STI, denominada *Intelligent Teaching Assistant* (ITA), cujo objetivo principal é oferecer assistência aos professores, reduzindo e automatizando tarefas repetitivas.

Ao consultar a bibliografia sobre os efeitos da interação promovida pelos sistemas entre professores e alunos, há o experimento de Raabe e Giraffa (2006). O estudo faz uso da teoria das Experiências de Aprendizagem Mediadas (EAM), de Feuerstein (1998), que inclui o professor como um dos agentes que interagem no ciclo de tutoração do aluno. Foi proposta uma arquitetura baseada em ITA, aplicada em turmas da disciplina de algoritmos, que possibilitou o auxílio ao professor na construção do diagnóstico do aluno. O estudo comprovou que, com a adoção da arquitetura, foi possível ofertar um acompanhamento individualizado aos alunos que apresentavam maior nível de dificuldade, tendo como consequência, a melhoria no desempenho dos alunos. Isso se deve ao fato de que as dificuldades iniciais de aprendizagem dos alunos foram detectadas mais cedo, possibilitando que o professor pudesse reorganizar suas ações.

Um aspecto negativo, porém esperado pelos pesquisadores, foi o aumento da demanda de trabalho do docente diante da necessidade de correção das questões resolvidas pelos alunos. Tal constatação não constitui entretanto, um problema para o presente trabalho, pois como a correção na FARMA-ALG é realizada de forma automática, esse aspecto

não é relevante. Verifica-se aqui que a interação do professor, de modo mais individual e personalizado sobre cada questão do aluno, pode contribuir para a melhoria do processo pedagógico. É portanto, essencial que se formalize uma metodologia que oriente o professor sobre como ele pode utilizar a FARMA-ALG como ferramenta que, além do seu propósito principal – oferecer retorno sobre o erro do aluno – também possa proporcionar assistência ao docente.

O tratamento de erros através de sua mediação em ferramentas de autoria tem como exemplos o projeto CARRIE de Marczal e Direne (2011), a ferramenta FARMA, de Marczal e Direne (2012) e posteriormente a ferramenta FARMA-ALG, de Kutzke e Direne (2014). O projeto CARRIE foi o precursor da FARMA, que por conseguinte, fundamenta-se como um dos pilares da FARMA-ALG. As três soluções têm como proposta a retroação ao estado de erro para sua mediação e promoção do aprendizado. É de extrema importância, além de saber da existência dessas soluções e de adotá-las, avaliar sua utilização, seus recursos, características e vantagens.

## **2.3 Avaliação de ferramentas para mediação de erros**

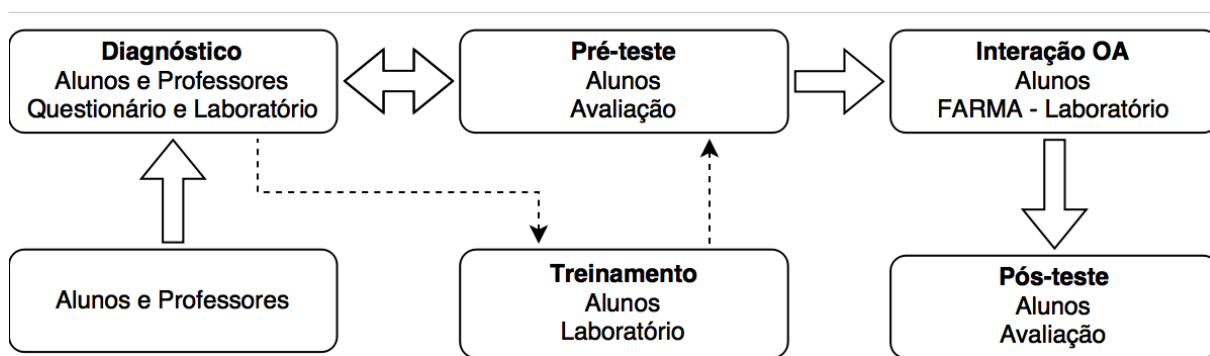
Ainda que os sistemas citados promovam a interação entre professores e alunos, eles carecem de estudos que nos permitam afirmar de que forma o resultado dessa interação contribui para ambos. O trabalho de Moura e Peres (2017) descreve a avaliação da ferramenta FARMA, envolvendo 260 alunos do ensino fundamental e médio da rede pública. O objetivo da pesquisa das autoras consistiu em analisar a utilização do mecanismo de retroação de erros e acertos por meio de encaminhamentos metodológicos. Os resultados demonstram que a ferramenta cria um ambiente favorável para que alunos questionem, expressem e colaborem.

As autoras também verificaram que o mecanismo de retroação possibilitou que a medição de erros e acertos auxiliaram efetivamente alunos e professores a identificar os erros cometidos e então atuar sobre eles. A conclusão tem embasamento em Weisz e Sanchez (2002), cuja prerrogativa de ensino-aprendizagem afirma que *“um olhar cuidadoso sobre o que o aluno errou pode ajudar o professor a descobrir o que ele tentou fazer”*. Sobre

o desempenho dos alunos com o uso da FARMA, as autoras concluíram que houve ganho e acredita que mais pesquisas devam ser realizadas envolvendo sistemas que promovam a retroação.

Cabe destaque aqui para o método que as autoras utilizaram para avaliar a FARMA, que se baseia em cinco atividades distintas: diagnóstico, treinamento, pré-teste, interação OA e pós-teste. A sequência das atividades é descrita na Figura 2.1.

Figura 2.1: Sequência das atividades de avaliação da FARMA



Fonte: (MOURA; PERES, 2017) Adaptado pelo autor.

As autoras explicam que na fase de Diagnóstico, são verificadas as habilidades dos alunos e professores para manuseio de recursos computacionais que serão disponibilizados. A atividade de Treinamento objetiva o nivelamento utilizando como parâmetro os resultados do Diagnóstico. O Pré-teste auxilia na verificação de conhecimentos prévios dos alunos sobre o conteúdo da disciplina de matemática. Na Interação OA o aluno tem seu primeiro contato com um OA, sendo divididos entre Grupo Controle (GC), que não terá acesso aos recursos de retroação e Grupo Experimental (GE), que terá amplo acesso aos recursos de retroação disponíveis na ferramenta. O Pós-teste avalia a efetividade do recurso de retroação aplicando um novo teste, com questões diferentes das do pré-teste, porém avaliando as mesmas habilidades.

## 2.4 Metodologias de uso para ferramentas interacionistas

É importante salientar que o desenvolvimento de um STI requer um esforço concentrado e uma diversidade de conhecimentos e técnicas. Vicari e Giraffa (2003 apud SILVA, 2006) afirmam que, em geral, esse desenvolvimento envolve uma equipe multidisciplinar, composta por profissionais da Informática, Pedagogia, Comunicação, Psicologia e outras áreas, visto que a Inteligência Artificial (IA) é uma área tipicamente interdisciplinar. Ainda que haja forte esforço da comunidade científica em apresentar soluções, estas se restringem às metodologias para construção dos componentes dos STIs, não havendo a dedicação sobre como o professor irá utilizar os recursos de forma efetiva. É importante construir uma estratégia que considere fatores diversos, como o nível de conhecimento do estudante, o domínio, a motivação e as características afetivas do mesmo (GIRAFFA; VICARI, 1999).

Para que uma ferramenta como a FARMA-ALG atinja seus objetivos, é necessário prioritariamente, que haja o maior nível de interação possível entre alunos e professores. É desejável, portanto, que seus usuários enfrentem o menor número de dificuldades durante seu uso, preferencialmente nenhuma. Verifica-se, todavia, que há escassa literatura disponível sobre o assunto e, quando existe, normalmente não se trata de uma documentação oficial da plataforma, mas estudos científicos de terceiros. São raros os estudos que orientam como os STIs devem ser utilizados, tanto técnica quanto pedagogicamente, no sentido de auxiliar seus usuários para que seu uso seja guiado do modo correto.

O trabalho de Garcia e Direne (2013) é de grande relevância pois traz como diferencial propostas para estratégias colaborativas em STIs. Os autores propõem uma metodologia generalista baseada em processos para a realização bem sucedida de sessões de aprendizagem, que são aplicáveis para professores e estudantes. Com base em seus estudos e considerando as necessidades detectadas, no presente trabalho foi desenvolvida uma metodologia de uso aplicável à FARMA-ALG cujos detalhes são detalhados na Seção 5.3.

Ainda que não tenha sido possível avaliar as características e vantagens da FARMA-ALG com a metodologia proposta por Moura e Peres (2017), principalmente sua utilização pelos alunos, no Capítulo 3 é descrita sua arquitetura e analisado o seu uso sob a perspectiva do docente. No caso do presente trabalho, é importante frisar que não houve tempo

hábil para a aplicação dessa metodologia de avaliação para a FARMA-ALG. É desejável e recomendável para trabalhos futuros, que seja efetuada uma ampla e revisada avaliação da FARMA-ALG utilizando a metodologia proposta pelas autoras.

## **CAPÍTULO 3**

### **FARMA-ALG - MEDIAÇÃO DO ERRO NO ENSINO DE ALGORITMOS**

Para compreender a organização e o fluxo de funcionamento da FARMA-ALG, é necessário que sua arquitetura seja analisada e avaliada. Somente após conhecer os detalhes de sua implementação, é possível estudar a interação entre alunos e professores nos registros históricos e então propor uma metodologia de utilização da FARMA-ALG pelos professores. Este capítulo apresenta a ferramenta, fundamentada no trabalho de Kutzke e Direne (2014). Na Seção 3.1 é apresentada uma breve descrição da FARMA-ALG. Na Seção 3.2 são abordados a arquitetura da ferramenta e o funcionamento dos mecanismos de armazenamento, recuperação, visualização e manipulação dos registros de respostas da ferramenta. Uma análise das características propostas pela ferramenta é apresentada na Seção 3.3.

#### **3.1 Descrição da FARMA-ALG**

A proposta central da FARMA-ALG é promover a mediação do erro no ensino de programação de computadores considerando uma abordagem sociointeracionista (KUTZKE; DIRENE, 2014). Sua implementação teve como base a ferramenta FARMA, um arcabouço que permite a construção de exercícios de conceitos matemáticos (MARCZAL et al., 2015). A escolha se justifica por ela apresentar um mecanismo eficiente para autoria de OAs e possuir um mecanismo de armazenamento automático de respostas. Além do recurso de armazenamento e recuperação das repostas dos alunos, que permite a retroação aos estados de erros, presentes na FARMA, a FARMA-ALG oferece um mecanismo de visualização e análise das respostas considerando os grafos de relações existentes entre vários registros.

### 3.2 Arquitetura da FARMA-ALG

Para explicar a arquitetura da FARMA-ALG, é necessário discorrer sobre o funcionamento da FARMA, cuja arquitetura foi a base para a implementação do arcabouço do sistema para auxílio à mediação do erro, proposto por Kutzke e Direne (2015). O esquema de funcionamento da FARMA-ALG foi herdado da FARMA e é apresentado na Figura 3.1. Em primeiro lugar, exercícios de matemática ou de disciplinas afins como física e química são produzidos e disponibilizados pelos professores para as turmas. A interação dos alunos com o sistema tem início quando eles efetuam suas matrículas nas turmas para que tenham acesso aos exercícios.

Na FARMA, quando o aluno submete uma resposta, ele é informado se ela está correta ou não e pode receber sugestões propostas pelo professor em caso de erro. Como cada resposta fica armazenada em um registro, tanto o aluno quanto o professor podem realizar a retroação ao exato estado errôneo e então analisar o contexto de um erro (MARCZAL, 2014). O principal diferencial do arcabouço que foi proposto na FARMA-ALG, conforme a Figura 3.1, é a possibilidade de ir além do armazenamento e recuperação de respostas por meio da retroação ao estado de erro. O docente pode então ter acesso a análises aprofundadas dos registros armazenados, considerando as relações existentes entre as respostas de diferentes alunos.

Para atender as necessidades inerentes ao ensino de programação, foi necessário efetuar algumas alterações no funcionamento da FARMA. A principal nova funcionalidade, adicionada à FARMA-ALG, foi a introdução de um sistema avaliador, similar ao BOCA<sup>1</sup>. A avaliação é feita pelo sistema ao comparar as respostas submetidas pelos alunos com os casos de testes previamente fornecidos pelo professor. Caso o aluno submeta uma resposta errada, ele pode receber comentários que o auxiliem na correta resolução do problema.

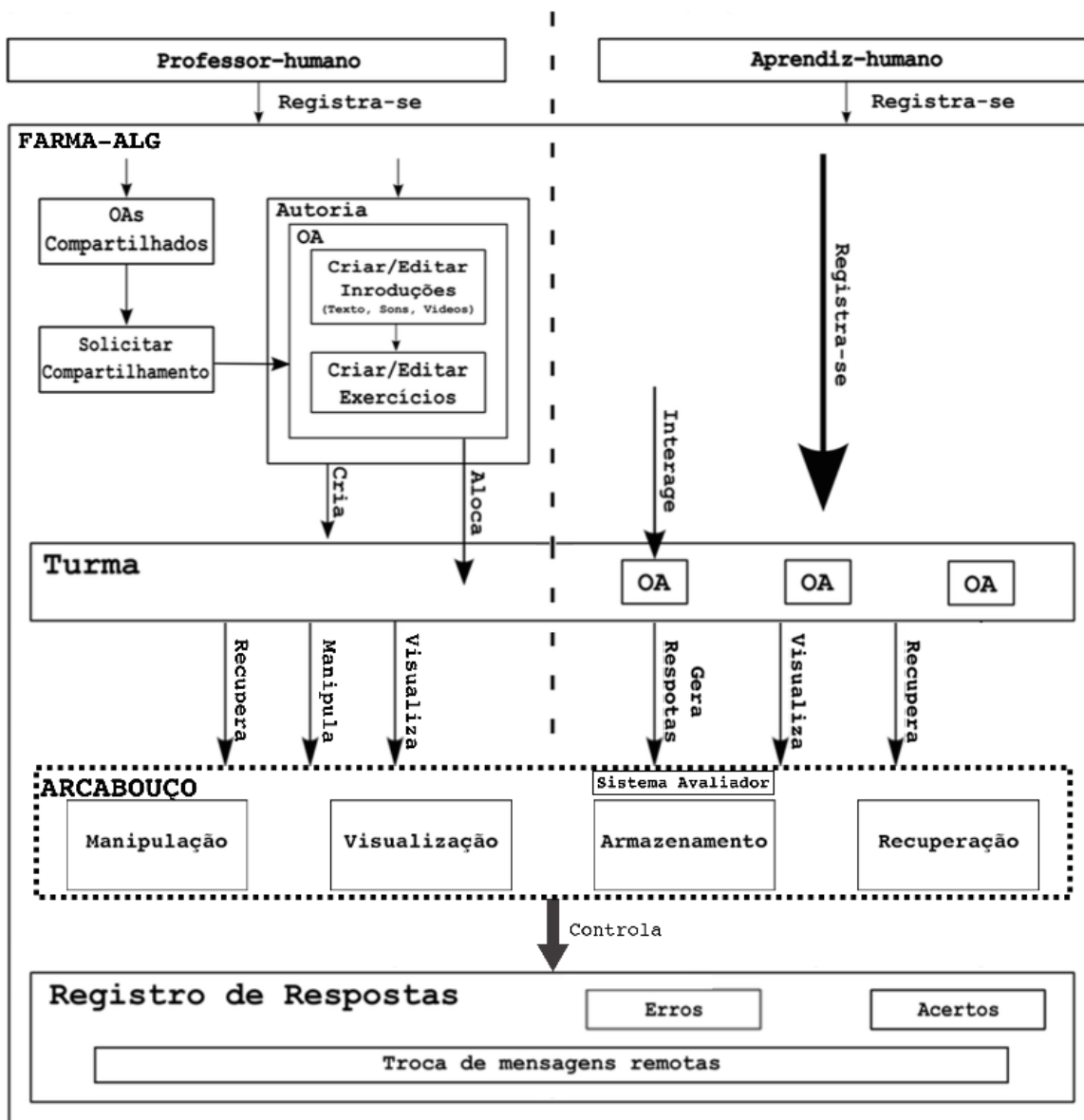
No quadro destacado em linhas pontilhadas na Figura 3.1, está o arcabouço implementado, que cria uma interface entre as ações dos usuários e os registros de respostas armazenados. É por meio dessa interface que as novas funcionalidades sobre a arquitetura da FARMA são adicionadas. O arcabouço contempla quatro módulos: armazenamento, recuperação,

---

<sup>1</sup> <<http://www.ime.usp.br/~cassio/boca/>>



Figura 3.1: Arquitetura da FARMA-ALG



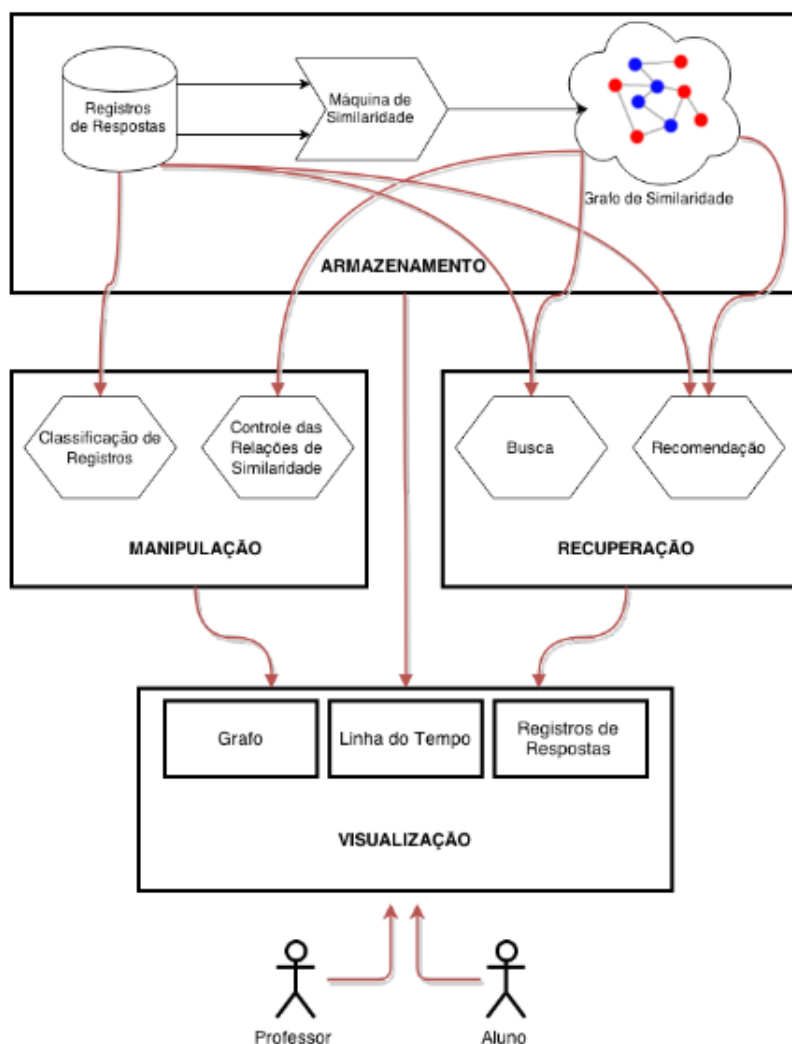
Fonte: (KUTZKE; DIRENE, 2015)

visualização e manipulação. Sua organização pode ser conferida na Figura 3.2 e suas características e principais funcionalidades são abordadas a seguir.

### 3.2.1 Armazenamento

O mecanismo de armazenamento automático de respostas da FARMA foi mantido na FARMA-ALG. Foram adicionadas duas novas funcionalidades: a máquina de similaridade e o grafo de similaridade.

Figura 3.2: Organização do arcabouço



Fonte: (KUTZKE; DIRENE, 2015)

A máquina de similaridade permite calcular o grau de semelhança entre duas respostas, de acordo com parâmetros definidos. São verificadas falhas de segmentação, erros de compilação, utilização de memória, tempo de execução e as semelhanças entre a saída obtida para cada conjunto de testes. Também são verificados os graus de semelhança entre os códigos-fonte das repostas e entre as saídas obtidas para cada caso de teste.

O grafo de similaridade é formado à medida em que os alunos vão submetendo suas respostas na FARMA-ALG. A máquina de similaridade procura por semelhanças em respostas de alunos que sejam da mesma turma para uma mesma questão. Caso não sejam encontradas semelhanças, é realizada uma nova comparação para a mesma questão,

entretanto, de turmas diferentes. Por fim, a resposta é comparada com outras questões, em busca de similaridades de código-fonte.

### 3.2.2 Recuperação

A recuperação de respostas, considerada pelos autores como um dos componentes basilares da solução proposta, é composta pelas funcionalidades de busca de registros e de recomendação. Na funcionalidade de busca, é disponibilizado ao professor e aluno um sistema de busca por palavras-chave, permitindo que ambos possam encontrar respostas de assuntos específicos. Dessa forma é possível recuperar dados que sejam relevantes para o processo de mediação do erro.

O sistema de busca é também capaz de recuperar trechos de código-fonte presentes nas respostas dos alunos, permitindo que o professor busque por construções comuns, como `array[min..max]`, por exemplo. Durante o procedimento de busca, o professor pode filtrar os resultados por intervalo de tempo, usuário, turma, objeto de aprendizagem, questão ou categoria.

A funcionalidade de recomendação tem como base a análise do grafo de similaridade das respostas. Ao realizar a análise, o sistema encontra registros que podem ser relevantes para um determinado grupo de alunos, sugerindo recomendações. O sistema então oferece a possibilidade do professor aceitar as recomendações e enviar comentários sobre elas aos alunos.

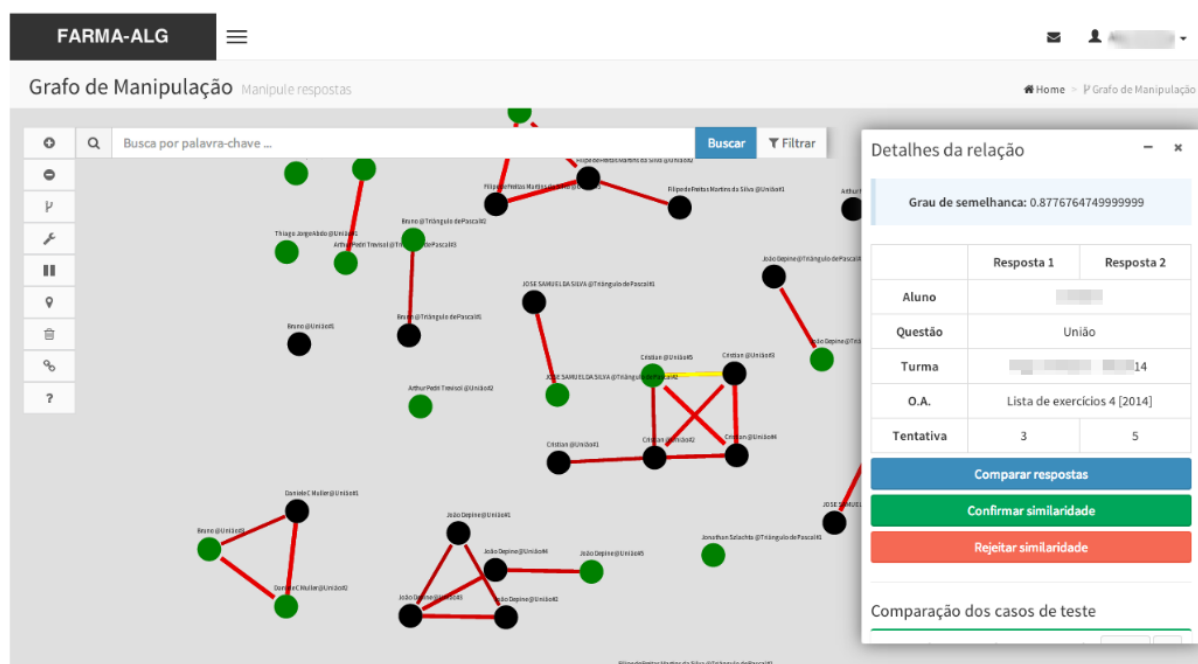
### 3.2.3 Visualização

Para que o professor possa manipular os registros armazenados na FARMA-ALG, é necessário que ele tenha ao seu dispor as ferramentas de visualização necessárias. Os seguintes modos estão disponíveis: grafo, linha do tempo, visualização de respostas e outras visualizações.

No sentido de auxiliar o professor na tarefa de manipulação dos registros, a funcionalidade de visualização gráfica do grafo de manipulação é um recurso de destaque. Ao visualizar o grafo, é possível observar características de uma única resposta bem como das relações entre

respostas diferentes. A Figura 3.3 mostra a tela de um grafo com as opções disponíveis em um menu vertical, tais como, adicionar ou remover respostas, adição de aresta, pausa do movimento do grafo, centralização e remoção de respostas.

Figura 3.3: Grafo de similaridade



Fonte: Autor

### 3.2.4 Manipulação

As respostas incorretas são apresentadas como vértices de cor preta e as respostas corretas, em vértices de cor verde. Ao selecionar a aresta de uma relação entre vértices, é apresentada uma visualização com os detalhes da relação. A partir dessa tela, o professor pode executar operações, como comparar as respostas e então confirmar ou rejeitar a similaridade entre elas.

Uma das principais funcionalidades da FARMA-ALG, segundo Kutzke e Direne (2014), é o agrupamento automático de respostas similares com a possibilidade de visualização e manipulação dos registros pelo professor. O professor tem a opção de usar a ferramenta de visualização de grafo de similaridade e então operar sobre os agrupamentos, apropriar-se das informações armazenadas e aplicar as intervenções que considerar necessárias. O sistema oferece um mecanismo de atribuição automática de relações entre respostas, realizada pela

máquina de semelhança. Porém, é permitido ao professor a manipulação dos registros, inserindo ou removendo relações entre elas, nos casos em que a máquina de semelhança não foi capaz de fazê-las automaticamente.

Quanto mais o professor interage com o sistema através de suas ações de manipulação dos registros de respostas, mais “treinada” a aplicação se tornará. A FARMA-ALG oferece ao professor duas formas distintas para que ele realize a manipulação dos registros. Ele pode operar diretamente o grafo de similaridade, o que facilita sua abstração e compreensão, ou acessar as respostas disponíveis em uma linha do tempo, de forma que ele pode analisar o contexto da formação dos conceitos do aluno. Assim, o docente contribui com subsídios para que a máquina de semelhança possa melhorar as generalizações automáticas das categorizações das respostas.

### 3.3 Análise da FARMA-ALG

As repostas submetidas pelos alunos são armazenadas automaticamente em um banco de dados *no-SQL* MongoDB. O módulo de armazenamento, mostrado na Figura 3.1, foi herdado da FARMA e é o responsável em coletar automaticamente todas as submissões de respostas. Para obter acesso à base de respostas, foi solicitado ao autor da ferramenta que efetuasse uma cópia completa contendo todos os registros das turmas que foram pesquisadas. A visualização e manipulação dos registros da base de respostas envolveu a criação de um banco de dados local que foi populado com os dados provenientes da base fornecida.

Segundo os autores da FARMA-ALG, a ferramenta permite criar e disponibilizar objetos de aprendizagem para o ensino de programação de computadores. Dessa forma, torna-se possível ao professor manipular, classificar e entender as relações entre as repostas de seus alunos. A ferramenta é disponibilizada via web, sua interface é leve, intuitiva e agradável, o que caracteriza, desde o acesso, a vocação sociointeracionista proposta.

### 3.3.1 Observações sobre o uso da interface

Ao acessar a FARMA-ALG<sup>2</sup>, a tela inicial oferece as opções de entrar como um usuário já cadastrado, registrar-se ou obter informações detalhadas sobre a ferramenta. Ainda na tela inicial, são mostradas miniaturas de alguns módulos específicos da FARMA-ALG, com breves explicações sobre as funcionalidades de cada uma delas, como manipulação das respostas, recomendações automáticas para análise de respostas relevantes e os diferentes tipos de visualizações disponíveis.

Os desenvolvedores se dedicaram aos detalhes de Interface Humano-Computador (IHC). Ao selecionar a opção **Entrar**, o usuário, que já deve estar cadastrado, será direcionado para a Figura 3.4 (a). Caso ainda não tenha efetuado o registro, é possível selecionar a opção **Registrar-se**, que aparece na parte inferior da mesma tela, sem necessidade de voltar à tela inicial. De forma similar, o mesmo ocorre se o usuário optar no início em **Registrar-se**. Nesta fase, ele tem a opção de escolher se o registro será efetuado como aluno ou professor. O procedimento de registro é bastante simplificado, solicitando apenas dados como nome completo, endereço eletrônico, se é aluno ou professor, a senha e sua confirmação. Caso o usuário tenha se equivocado e já tenha efetuado o seu registro, ele pode facilmente acessar a tela correspondente através da opção **Entrar**, disponibilizada na parte inferior da Figura 3.4 (b). Ao acessar o sistema pela primeira vez, é apresentada uma mensagem de boas vindas ao usuário, com explicações sobre as opções disponíveis, conforme a Figura 3.5.


Após a leitura, o usuário pode optar em fechar o aviso ou selecionar a opção para que a mensagem não seja exibida novamente, otimizando a visualização de outros itens. A tela inicial da FARMA-ALG é bastante simples e intuitiva contando com um painel de mensagens relevantes para o professor, que exibe itens como número de respostas corretas e incorretas, quantidade de turmas e alunos, comentários e mensagens, as últimas respostas das turmas e as recomendações feitas automaticamente pelo sistema. Na lateral esquerda há um completo menu que dá acesso a todas as funcionalidades disponíveis.

A interface foi desenvolvida considerando que o acesso pode ser efetuado em qualquer

---

<sup>2</sup> <<http://farma-alg.com.br/>>

Figura 3.4: Opções para Entrar ou Registrar



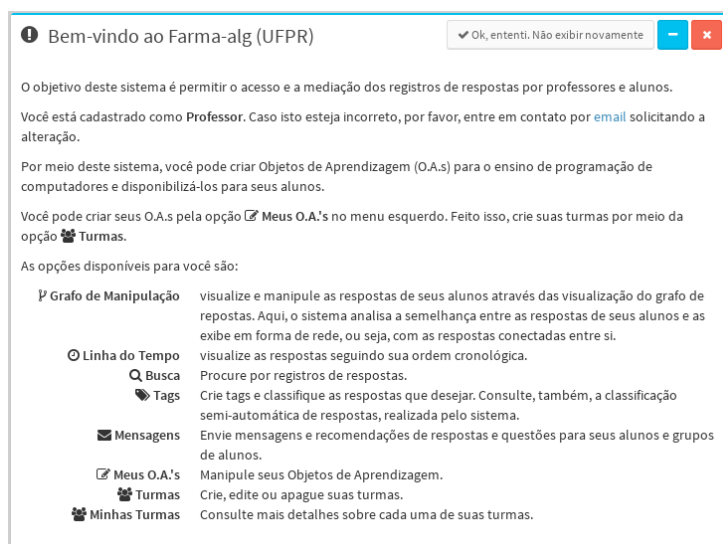
(a) Entrar



(b) Registrar

Fonte: Autor

Figura 3.5: Mensagem de boas vindas



**Bem-vindo ao Farma-alg (UFPR)** [Ok, entendi. Não exibir novamente] [Fechar]

O objetivo deste sistema é permitir o acesso e a mediação dos registros de respostas por professores e alunos.

Você está cadastrado como **Professor**. Caso isto esteja incorreto, por favor, entre em contato por [email](#) solicitando a alteração.

Por meio deste sistema, você pode criar Objetos de Aprendizagem (O.A.s) para o ensino de programação de computadores e disponibilizá-los para seus alunos.

Você pode criar seus O.A.s pela opção ☒ **Meus O.A.'s** no menu esquerdo. Feito isso, crie suas turmas por meio da opção **Turmas**.

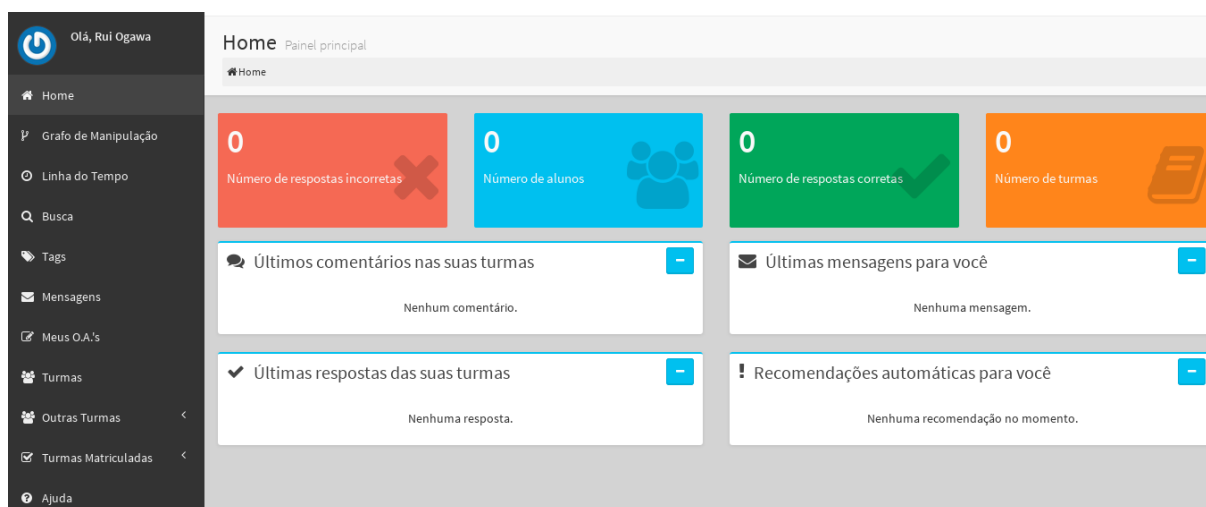
As opções disponíveis para você são:

|  |   |
|--|---|
| <b>Grafo de Manipulação</b>                            | visualize e manipule as respostas de seus alunos através da visualização do grafo de repostas. Aqui, o sistema analisa a semelhança entre as respostas de seus alunos e as exibe em forma de rede, ou seja, com as respostas conectadas entre si. |
| <input type="radio"/> <b>Linha do Tempo</b>            | visualize as respostas seguindo sua ordem cronológica.  |
| <b>Busca</b>   | Procure por registros de respostas.   |
| <b>Tags</b>  | Crie tags e classifique as respostas que desejar. Consulte, também, a classificação semi-automática de repostas, realizada pelo sistema.  |
| <b>Mensagens</b>                                       | Envie mensagens e recomendações de repostas e questões para seus alunos e grupos de alunos.   |
| <input checked="" type="checkbox"/> <b>Meus O.A.'s</b> | Manipule seus Objetos de Aprendizagem.  |
| <b>Turmas</b>  | Crie, edite ou apague suas turmas.  |
| <b>Minhas Turmas</b>                                   | Consulte mais detalhes sobre cada uma de suas turmas.   |

Fonte: Autor

tipo de dispositivo e, portanto, conta com recurso de responsividade, permitindo que seus elementos gráficos e menus se ajustem automaticamente à qualquer resolução de tela. A

Figura 3.6: Tela principal da FARMA-ALG



Fonte: Autor

Figura 3.7 mostra o sistema sendo acessado por um telefone celular, com o menu oculto e exibido, respectivamente.

### 3.3.2 Observações sobre a criação de OAs

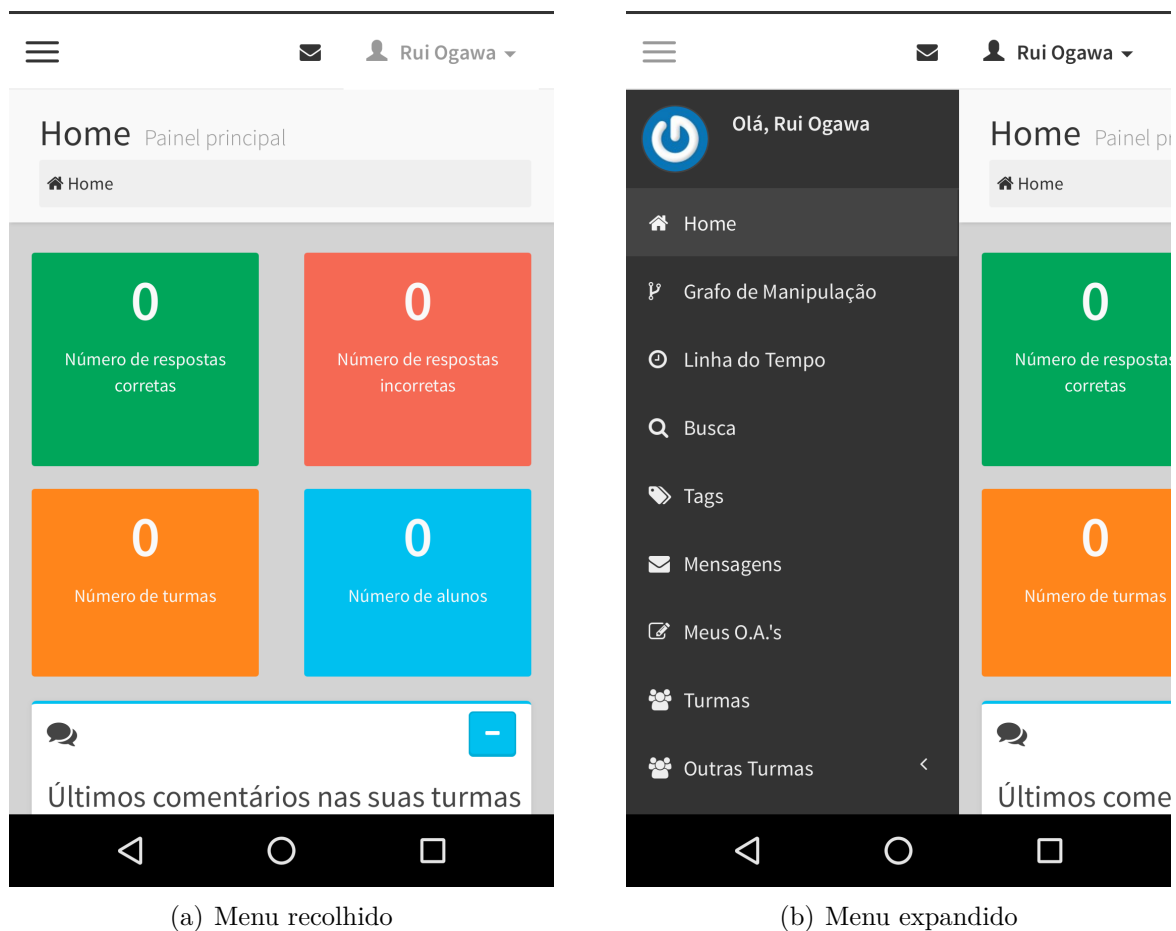
A FARMA-ALG provê funcionalidades para criação, edição e compartilhamento de OAs entre as turmas. A tarefa de criar um OA revelou-se ser de certa facilidade, uma vez que a ferramenta é provida de um editor que orienta o professor para que, em poucos e intuitivos passos, seja possível criar sua introdução e os exercícios relacionados. Na Figura 3.8, temos o exemplo de criação de um OA para a fórmula de Bhaskara.

O professor tem a possibilidade de, imediatamente, disponibilizar o OA para seus alunos, marcando a opção Publicar. Entretanto, apenas a criação de um OA não contempla itens fundamentais, como a introdução, os exercícios e as questões propostas. Portanto, a disponibilização do OA aos alunos só faz sentido após a criação de todos os itens necessários. Quando um OA ainda não está publicado, há uma marca de cor vermelha no lado esquerdo, indicando que está indisponível para os alunos.

Após a criação do OA, o professor tem a possibilidade de criar as introduções que julgar necessárias e os exercícios, conforme é mostrado na Figura 3.9. A criação de uma introdução para um OA é realizada de forma simples, com o auxílio de um editor embutido na FARMA-ALG. O objetivo da introdução é prover subsídios científicos no OA, para



Figura 3.7: Interface responsiva



Fonte: Autor

Figura 3.8: Criação de um novo OA

Objetos de Aprendizagem / novo

### Novo objeto de aprendizagem para o OA

Nome \*

Descrição \*

☒ Publicar (Disponibilizar para alunos)

Fonte: Autor

Figura 3.9: OA disponível



Fonte: Autor

que o aluno tenha uma boa compreensão sobre os temas que são abordados nos exercícios. Como exemplo, a Figura 3.10 mostra a contextualização da fórmula de Bhaskara. Assim como na criação do OA, a introdução também pode ser disponibilizada imediatamente ou posteriormente aos alunos.

Figura 3.10: Introdução de um OA

Título \*

Contextualização

Conteúdo \*

Código-Fonte

✂

📄

📁

📁

📁

↶

↷

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

**B**

*I*

U

~~S~~

$\times_2$

$\times^2$

$I_x$

☰

☰

☰

☰

☰

☰

☰

☰

☰

☰

☰

☰

☰

☰

☰

☰

☰

☰

☰

☰

☰

☰

☰

☰

Estilo

Normal

Fonte

Ta...

A

A

🔍

🔍

?

🔍

🔍

🔍

🔍

🔍

🔍

🔍

🔍

O nome **Fórmula de Bhaskara** foi dada em homenagem ao matemático **Bhaskara Akaria**, considerado o mais importante matemático indiano do século XII.

A fórmula de Bhaskara é principalmente usada para resolver equações quadráticas de fórmula geral  $ax^2+bx+c=0$ , com coeficientes reais, com  $a \neq 0$  e é dada por

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

chamamos de discriminante:  $\Delta = b^2 - 4ac$

body p

☐ Publicar (Disponibilizar para alunos)

Fonte: Autor

Após a formulação da introdução, o professor pode criar uma ou mais listas de exercícios, que conterão as questões. Dessa forma, a disponibilização do OA aos alunos só faz sentido após a criação de todos os exercícios e das questões. Após a criação dos exercícios, a formulação de questões também se mostrou simples, podendo ser efetuada em poucos passos. Em todos os passos é disponibilizado um editor embutido no formulário, com recursos de formatação no estilo WYSIWYG<sup>3</sup>. Ao criar uma ou mais questões para um determinado exercício, é solicitado ao autor que marque a opção que determina em qual

---

<sup>3</sup> *What You See Is What You Get.*



Figura 3.12: Casos de teste para raízes de uma equação de segundo grau

**Questões** [Criar nova questão](#)

Crie e edite questões.  
Lembre-se de adicionar pelo menos um caso de teste para cada questão.

**✚ Raízes de uma equação do segundo grau** [+ Detalhes](#) | [✓ Casos de teste](#) | [✎ Editar](#) | [🗑 Remover](#)  
Criado em Atualizado em

**Casos de teste da Questão Raízes de uma equação do segundo grau** [Criar novo caso de teste](#)

|                  |   |
|------------------|---|
| Sem raízes reais | <a href="#">+ Detalhes</a> <a href="#">✎ Editar</a> <a href="#">🗑 Remover</a> |
| Uma raiz         | <a href="#">+ Detalhes</a> <a href="#">✎ Editar</a> <a href="#">🗑 Remover</a> |
| Duas raízes      | <a href="#">+ Detalhes</a> <a href="#">✎ Editar</a> <a href="#">🗑 Remover</a> |

Fonte: Autor

dos casos de teste que demanda mais tempo e atenção do professor, pois as alternativas devem ser corretas e precisas, para que a resposta do aluno seja comparada com a saída de cada caso de teste.

Figura 3.13: Caso de teste para duas raízes

**Duas raízes**

**Descrição:**  
Quando houver duas raízes reais imprima:  
**A primeira raiz é X**  
**A segunda raiz é Y**  
onde X e Y são os valor das raízes.

**Dica:**  
**Apresentar a dica após 1 tentativa(s) incorreta(s)**  
**Tempo máximo de execução: 1 segundo(s)**

⚠ Este caso de teste irá ignorar diferenças de tabulação  
⚠ Este caso de teste irá apresentar entradas e saídas para os alunos

**Entrada**

```
10
25
10
```

**Saída**

```
A primeira raiz é: -0.5
A segunda raiz é: -2.0
```

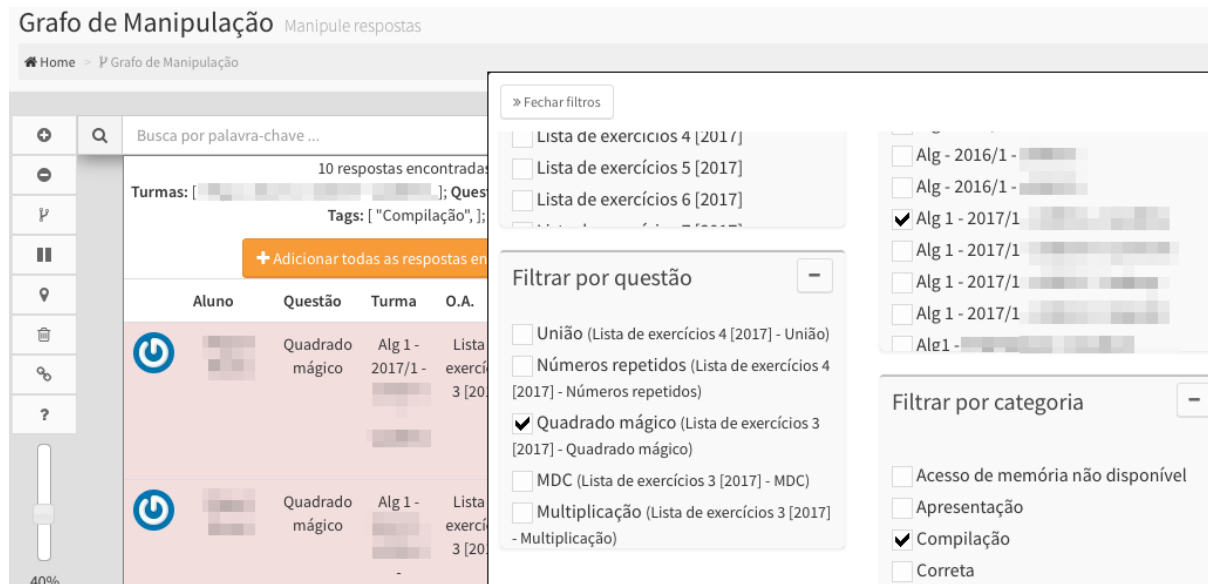
Fonte: Autor

### 3.3.3 Observações sobre o grafo de manipulação de respostas

A utilização do grafo de manipulação de respostas é simples e provida de muitos recursos para busca, filtragem e manipulação das respostas que aparecem. Através da visualização do grafo, foi possível analisar as relações entre as respostas, o que constitui recurso pedagógico para auxílio ao professor, considerando que este terá subsídios para o seu fazer pedagógico. Estes recursos, aliados à possibilidade de envio de mensagens e comentários aos alunos, corrobora para o sucesso da proposta sociointeracionista da ferramenta.

O recurso de grafo de manipulação permite efetuar buscas por palavras-chave ou por meio de filtros. É possível criar filtros diversos, podendo selecionar por listas de exercícios, turmas, questões e tipos de erros. Na Figura 3.14 foram encontradas 10 respostas similares que contém erros de compilação de uma determinada turma para a questão intitulada “Quadrado mágico”.

Figura 3.14: Exemplo de busca no grafo de manipulação

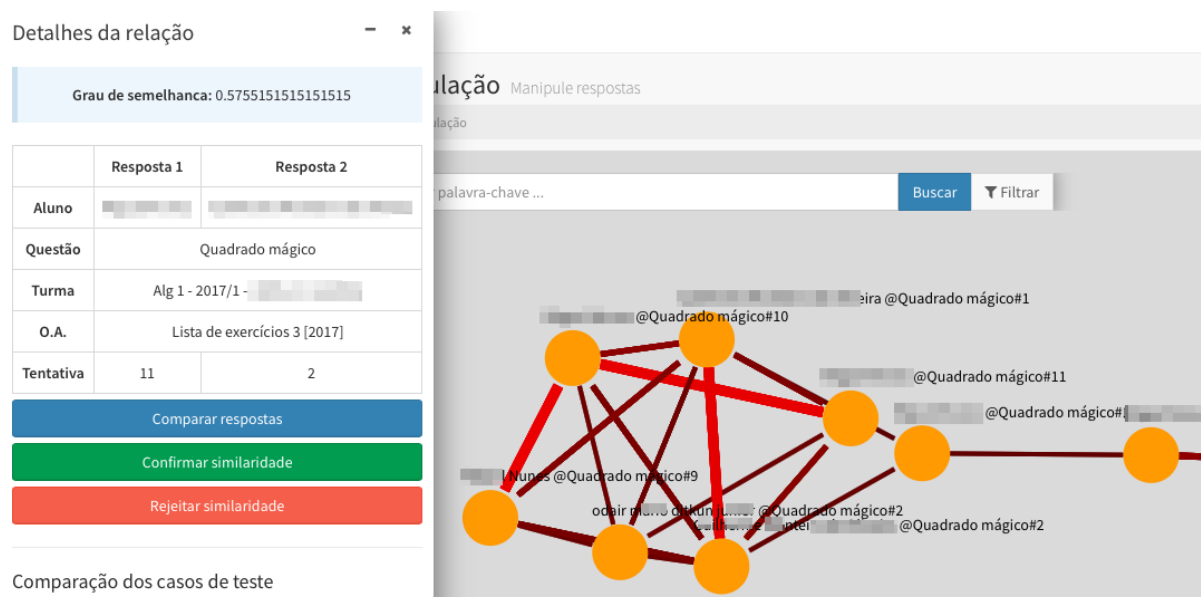


Fonte: Autor

Após a busca, o procedimento para adicionar as respostas encontradas ao grafo é simples, bastando clicar no botão “Adicionar todas as respostas encontradas”. A parte central da tela é preenchida com os vértices, que correspondem às respostas, interligados por arestas, que correspondem ao grau de similaridade entre as respostas. A Figura 3.15

mostra que, ao selecionar uma aresta com um clique duplo, uma janela exibe os detalhes da relação entre as duas respostas, informando o grau de semelhança, os nomes dos alunos, a questão, a turma selecionada, lista de exercícios e a quantidade de tentativas de cada aluno. Também é possível comparar os códigos das respostas lado a lado, bem como confirmar ou rejeitar manualmente a similaridade.

Figura 3.15: Resultado da busca no grafo de manipulação



Fonte: Autor

### 3.4 Considerações para melhoria da ferramenta

Na FARMA-ALG, as questões são avaliadas através de padrões de entrada e saída, também chamados de casos de teste, que devem ser manualmente construídos e fornecidos pelo professor. Caso o professor julgue necessário, vários casos de teste podem ser definidos para uma mesma questão. Para determinar se uma resposta está correta, verifica-se se todas as saídas obtidas pelo código do aluno, para cada caso de teste, estão corretas, ou seja, se são iguais às saídas registradas pelo professor no momento de criação dos casos de teste. A resposta do aluno é então testada para todos os casos definidos pelo professor e para cada um, tem seu resultado armazenado.

A precisão da correção de uma questão de um determinado exercício de algoritmo

está intrinsecamente ligada à quantidade e à qualidade dos casos de teste, que devem ser elaborados, testados e cadastrados na ferramenta manualmente, pelo professor. O sistema não provê ainda a possibilidade de receber uma solução de referência (correta) do professor. Segundo os autores da ferramenta, as únicas comparações que ocorrem são entre as saídas obtidas pelas respostas dos alunos com as saídas previstas nos casos de teste elaborados pelo professor e entre a similaridade das respostas dos alunos.

Ao avaliar essas funcionalidades e em discussão com os autores da FARMA-ALG, foi identificada a necessidade de três novas funcionalidades que contribuiriam para a melhoria da ferramenta. A primeira refere-se à possibilidade do professor enviar uma solução de referência (código-fonte) correta. A segunda funcionalidade receberia o código-fonte de referência submetido e criaria automaticamente todos os possíveis casos de teste para a questão, tirando do professor grande parte do trabalho manual atualmente necessário para a criação dos casos de teste. Finalmente, a última nova funcionalidade melhoraria o método de correção pela adição da correção do código submetido pelo aluno através de ferramentas de análise de código-fonte.

O aspecto documental da FARMA-ALG necessita de melhorias, pois na verdade, praticamente inexistente. Trata-se não somente de documentação sobre a modelagem e codificação da plataforma, mas também e principalmente, a respeito de como professores e alunos devem proceder para extrair o máximo dos benefícios oferecidos. Dentre os principais recursos, destaca-se a possibilidade da remediação do erro por meio da interação entre professores e alunos, via comentários e mensagens. A FARMA-ALG não conta, até o presente momento, com uma documentação que guie seus usuários para que o maior nível de interação seja atingido. Nesse sentido, é apresentada uma Metodologia com Enfoque na Interação, no Capítulo 5. Não basta, todavia, avaliar apenas as características de interface e usabilidade de uma ferramenta, como foi descrito nas seções anteriores, para que seja possível concluir sua efetividade. É necessário avaliar o quanto a FARMA-ALG tem sido capaz de atingir o seu principal objetivo, que é o de remediar o erro do aluno em questões de algoritmos pela interação com o professor. A efetividade da promoção da interação, como proposta da FARMA-ALG para remediação do erro, é avaliada no Capítulo 4.

### 3.5 Considerações finais sobre a FARMA-ALG

Neste capítulo, foram estudados aspectos sobre a arquitetura da FARMA-ALG e efetuadas observações sobre suas principais funcionalidades. Os estudos e observações foram necessários e de inequívoca utilidade para que tivéssemos a melhor compreensão possível sobre o contexto para o qual a ferramenta foi concebida, suas funcionalidades e propósitos. Compreender o funcionamento da plataforma contribuiu para o estudo de sua efetividade bem como serviu como base para a elaboração da metodologia proposta neste trabalho.

A FARMA-ALG foi analisada em seus aspectos de uso da interface, que mostrou ser simples, concisa e organizada, provendo fácil acesso às funcionalidades desejadas. Sobre a criação de OAs, as observações mostraram que o professor é contemplado com um método relativamente simples para a confecção dos exercícios e questões. A principal dificuldade encontrada reside na necessidade em criar manualmente todos os casos de teste para cada questão, atividade que consome considerável tempo do professor.

As observações sobre a utilização do grafo de manipulação revelaram que esse recurso da ferramenta permite que o professor tenha subsídios suficientes para tomada de decisão sobre o reforço que deve ministrar sobre um determinado aspecto do algoritmo no qual os alunos estejam com dificuldades. Ao final, são elencadas algumas considerações para a melhoria da FARMA-ALG, principalmente no que concerne a automatização para a criação de casos de teste.



## **CAPÍTULO 4**

### **ESTUDOS DA EFETIVIDADE DA INTERAÇÃO**

Foi definida como objeto desta dissertação, a verificação da efetividade que a utilização da ferramenta FARMA-ALG produz no processo de ensino-aprendizagem das disciplinas de algoritmos, em especial considerando as características sociointeracionistas da solução. Uma vez definidos os objetivos da dissertação e compreendido o funcionamento da FARMA-ALG, iniciou-se a pesquisa bibliográfica para definir os métodos e técnicas que seriam utilizados para o estudo. Para estudar a efetividade da ferramenta, foram efetuadas duas ações principais: a avaliação geral da distribuição das respostas submetidas pelos alunos e a análise crítica e estatística das interações. Os métodos que foram utilizados são abordados em detalhes nas próximas seções.

O universo da pesquisa é mostrado na Seção 4.2. Cada procedimento de avaliação é detalhado nas seções subsequentes, explicando seus objetivos e métodos utilizados para coleta e análise dos dados e dos resultados esperados. A influência da FARMA-ALG nos resultados é avaliada na Seção 4.3. Na Seção 4.4 é feita uma análise crítica da interação entre professores e alunos por meio das mensagens e comentários enviados. Finalmente, na Seção 4.5 é realizada a análise estatística descritiva e inferencial por meio de testes e técnicas específicas.

#### **4.1 Objetivos dos estudos da efetividade da interação**

Conforme foi visto anteriormente, o objetivo deste trabalho é propor uma metodologia direcionada ao aumento da interação entre os professores e alunos que usam a FARMA-ALG, avaliando como e quanto a interação do docente com o aluno pode contribuir para a melhoria deste processo de ensino-aprendizagem. Para o desenvolvimento da pesquisa, foram previstas as seguintes ações, definidas nos objetivos e reproduzidas aqui novamente por conveniência:

- estudar a arquitetura da FARMA-ALG;
- estudar a interação que ocorreu entre os alunos e professores nos registros históricos da FARMA-ALG, por meio de análise estatística descritiva e inferencial das mensagens de interações entre professores e alunos;
- propor uma metodologia de utilização da FARMA-ALG pelos professores, para que os objetivos interacionistas sejam alcançados, melhorando os resultados e o processo de ensino-aprendizagem por meio do aumento quantitativo e qualitativo de interações.

## 4.2 Universo da pesquisa

Foram analisados na presente pesquisa, os dados das respostas de 229 alunos de sete turmas da disciplina de Algoritmos e Estrutura de Dados I, de duas universidades, que submeteram 8.887 respostas, coletadas entre 2015 e 2016. As respostas foram coletadas de três turmas do ano de 2015, totalizando 118 alunos de duas universidades que submeteram 4.854 respostas para questões aplicadas por dois professores. Quatro turmas são do ano de 2016, com 105 alunos de uma única universidade, que geraram 4.033 respostas para questões aplicadas por quatro professores. Todas as turmas cujas respostas foram coletadas são da fase inicial dos cursos. O objetivo foi avaliar a maior quantidade possível de respostas e interações entre professores e alunos que utilizaram a FARMA-ALG. A Tabela 4.1 mostra a distribuição dos componentes da turma de 2015.

Tabela 4.1: Turmas analisadas

| Universidade    | Turma    | Professor     | Alunos     | Respostas       |
|-----------------|----------|---------------|------------|-----------------|
| U1              | U1A-2015 | A             | 63         | 3.695           |
| U2              | U2A-2015 | B             | 13         | 365             |
| U2              | U2B-2015 | B             | 42         | 794             |
| U1              | U2A-2016 | A             | 28         | 1.266           |
| U1              | U2B-2016 | C             | 22         | 879             |
| U1              | U2C-2016 | D             | 34         | 1.206           |
| U1              | U2D-2016 | E             | 21         | 682             |
| 2 Universidades | 7 Turmas | 5 Professores | 229 Alunos | 8.887 Respostas |

A análise dos dados foi efetuada *a posteriori*, considerando que, quando da coleta do dados, o objetivo era somente a aprendizagem, não a observação na FARMA-ALG.

### 4.3 Observações da influência da FARMA-ALG nos resultados de aprendizagem

A utilização da FARMA-ALG tem como proposta auxiliar professores e alunos ao propiciar a mediação do erro em uma abordagem sociointeracionista. Somente será possível afirmar se a proposta obteve êxito se for avaliado como o uso da ferramenta influencia nos resultados de aprendizagem. A influência pode acontecer de diversas formas, como o aumento na quantidade de respostas corretas, redução do número de tentativas e diminuição do tempo dispendido para resolução de questões, melhoria das notas dos alunos e no aumento da interação entre alunos e professores.

Foram analisadas relações como percentuais de respostas corretas e incorretas, respostas que receberam intervenções através de comentários do professor, notas dos alunos e quantidade de tentativas. Por meio dessa análise, objetiva-se detectar comportamentos que possam evidenciar de que forma e em que nível o uso da FARMA-ALG influencia nos resultados. Com base nessa avaliação obteve-se informações suficientes para analisar a efetividade da FARMA-ALG como ferramenta de apoio à prática docente no ensino de algoritmos.

Procurou-se saber, por exemplo, em quais tipos de questões os alunos encontram maior ou menor grau de dificuldade e quais são os erros mais comuns. Um item destacado foi a análise da relação entre as respostas que receberam algum tipo de intervenção por parte do professor e quantidade de tentativas submetidas por questão. É apresentada a seguir a metodologia utilizada e os demais procedimentos que subsidiaram esta fase da pesquisa nas próximas seções.

### 4.3.1 Atividades para avaliação da influência da FARMA-ALG nos resultados

Os dados gerados pela FARMA-ALG foram utilizados para analisar estatisticamente o desempenho e progresso dos alunos. Uma análise preliminar dos dados, usando os recursos de consultas do Sistema Gerenciador de Banco de Dados (SGBD) MySQL<sup>1</sup>, permitiu ter um panorama geral sobre os dados. Foi possível verificar a distribuição das respostas às questões entre as turmas, quantidade de respostas corretas e incorretas, maior incidência de determinado tipo de erro, entre outros. Com base nessas comparações, é possível detectar possíveis tendências, como alguma dificuldade apresentada pelos alunos em determinada questão.

Para planejar como a análise dos dados seria realizada, foi necessário compreender a organização e relação entre as coleções mais relevantes para a pesquisa, sendo necessário portanto compreender a estrutura da base de dados que armazena as respostas na FARMA-ALG. A ferramenta não conta, entretanto, com uma documentação que explique aspectos computacionais, como, por exemplo, o banco de dados. Compreender os relacionamentos entre as tabelas consiste em tarefa fundamental para a consecução do presente trabalho. Todavia, como o SGBD adotado pela FARMA-ALG é o MongoDB<sup>2</sup>, baseado em coleções e não em tabelas, foi necessário desvendar os relacionamentos intrínsecos, pois não foi encontrada nenhuma documentação que mostrasse o Modelo Entidade Relacionamento (MER) e o Diagrama Entidade Relacionamento (DER). Portanto, para que fosse possível encontrar e listar os relacionamentos foi utilizada a ferramenta DBSchema<sup>3</sup>, um software modelador de banco de dados, compatível com o MongoDB. A Figura 4.1 mostra o diagrama, elaborado para este trabalho, das principais coleções e seus relacionamentos.

Uma vez conhecidos os relacionamentos entre as coleções, foi possível compreender o fluxo das informações entre elas para que fossem construídos os conjuntos de dados necessários para os processos de análise. Novamente, não foi possível encontrar uma

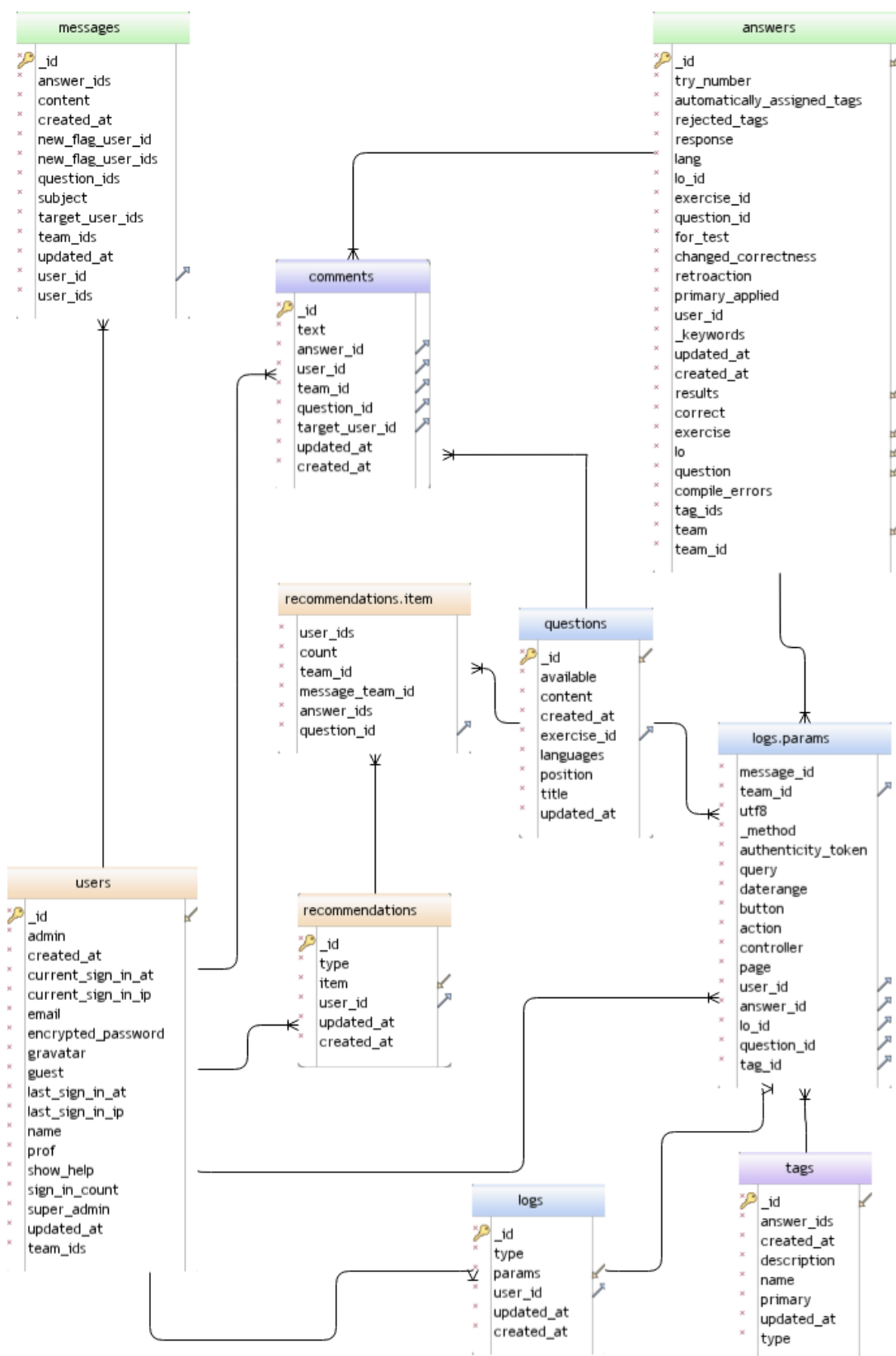
---

<sup>1</sup> <<https://www.mysql.com>>

<sup>2</sup> <<https://www.mongodb.com>>

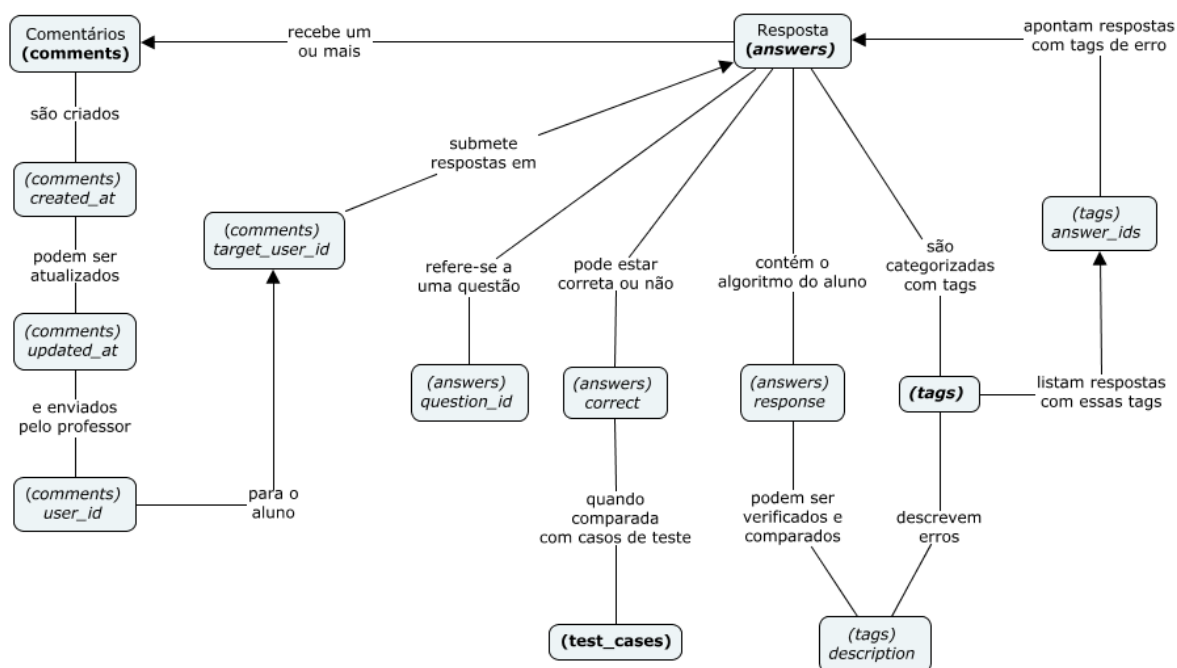
<sup>3</sup> <<http://www.dbschema.com/>>

Figura 4.1: Diagramas no DBSchema



documentação própria da FARMA-ALG que contribuisse para o entendimento de como fluxo ocorre. Com o objetivo de elaborar o fluxo, o uso da ferramenta de mapa conceitual CmapTools<sup>4</sup> demonstrou-se ser de grande ajuda para melhor visualização. A Figura 4.2 ilustra um mapa conceitual, construído neste trabalho de mestrado, que permitiu compreender melhor as ações necessárias.

Figura 4.2: Fluxo de respostas e comentários na FARMA-ALG



Houve alguns problemas relacionados com o manuseio da base de respostas, relatados a seguir. Como o MongoDB não é relacional, houve a necessidade de migração para um SGBD relacional, no caso o MySQL, para que fosse possível realizar consultas cruzadas. Os procedimentos para exportação podem ser vistos no Código 6.1. Durante a migração para o MySQL, pelo fato de que o MongoDB utiliza o conceito de coleções de documentos ao invés de tabelas, nos casos em que havia documentos aninhados, o procedimento padrão de exportação para CSV<sup>5</sup>, um formato utilizado para converter dados entre planilhas de cálculo, fez com que fossem criadas tabelas com várias colunas durante o processo de importação. Esse problema foi solucionado refazendo a exportação de modo que as coleções

<sup>4</sup> <<http://cmap.ihmc.us/cmaptools/>>

<sup>5</sup> <<https://tools.ietf.org/html/rfc4180>>

aninhadas fossem colapsadas. Campos do tipo `text`, que armazenam informações com formatação HTML desnecessária, bem como datas em campos `varchar`, dificultavam a importações, tendo sido requerida a sanitização de alguns campos e conversão de tipos.

Após ter acesso à base gerada pela FARMA-ALG durante sua utilização com as turmas, o próximo passo consistiu-se na instalação do MongoDB e de um gerenciador gráfico para auxiliar na visualização das coleções, no caso o MongoChef<sup>6</sup>. Uma vez identificadas as coleções consideradas relevantes para a pesquisa, elas foram selecionadas e exportadas para o formato CSV, para que pudessem ser importadas no MySQL. Depois que todas as coleções foram exportadas do MongoDB, elas foram importadas para a base criada no MySQL. Após a importação, foram criadas 15 tabelas correspondentes no MySQL, conforme Tabela 4.2. Os códigos SQL utilizados podem ser vistos no Código 6.2.

### 4.3.2 Resultados da avaliação da influência da FARMA-ALG nas respostas

Foram elaboradas 23 questões, elencadas na Tabela 4.3 e distribuídas em sete listas de exercícios, que foram disponibilizadas para três turmas de 2015 e para quatro turmas de 2016. A lista completa, com todas as questões disponibilizadas, incluindo seus enunciados, pode ser verificada no Tabela 6.1.

São apresentadas algumas estatísticas sobre os dados de respostas submetidas por alunos dessas turmas que utilizaram a FARMA-ALG. Foram analisadas as relações entre respostas corretas e incorretas, as classificações de erros e distribuição de tentativas de acordo com as listas de exercícios do conjunto de dados das duas turmas, comparando os resultados dos dados obtidos em 2015 e 2016. Todas as questões com suas respectivas quantidades de respostas e distribuição entre as listas de exercícios podem ser verificadas na Tabela 4.4.

Nas turmas de 2015, do total de 4.867 respostas submetidas, 1.249 (25,66%) eram corretas e 3.618 (74,34%) eram incorretas. Em 2016, 4.022 respostas foram registradas. Destas, 1.330 (33,07%) estavam corretas e 2.692 (66,93%) estavam incorretas. Ainda que

---

<sup>6</sup> <<http://3t.io/mongochef/>>

Tabela 4.2: Tabelas importadas para o MySQL

| <b>Tabela</b>   | <b>Descrição</b>   | <b>Registros<br/>2015</b> | <b>Registros<br/>2016</b> | <b>Total</b> |
|-----------------|--|---------------------------|---------------------------|--------------|
| answers         | Cada resposta, de cada aluno, para uma determinada questão         | 4.854                     | 4.033                     | 8.887        |
| comments        | Comentários enviados pelos professores para alunos                 | 80                        | 2                         | 82           |
| connections     | Indica o grau de similaridade entre respostas                      | 35.278                    | 33.778                    | 69.056       |
| last_answers    | Registro das últimas respostas submetidas                          | 1.033                     | 1.576                     | 2.609        |
| logs            | Log de eventos do sistema  | 9.347                     | 8.537                     | 17.884       |
| los             | Listas de exercícios   | 8                         | 8                         | 8            |
| messages        | Mensagens enviadas pelos professores para turma ou alunos          | 21                        | 2                         | 23           |
| progresses      | Indica o progresso de cada aluno para cada questão                 | 1.035                     | 1.586                     | 2.621        |
| questions       | Questões de algoritmos   | 23                        | 23                        | 23           |
| recommendations | Recomendações  | 12                        | 12                        | 24           |
| statistics      | Estatísticas de acordo com tentativas corretas, incorretas e erros | 81                        | 92                        | 173          |
| tags            | Categorias de tipos de erros esperados                             | 12                        | 12                        | 12           |
| teams           | As Turmas da disciplina de Algoritmos                              | 3                         | 4                         | 7            |
| test_cases      | Casos de testes  | 44                        | 44                        | 44           |
| users           | Usuários, incluindo alunos e professores                           | 129                       | 146                       | 275          |

a quantidade de acertos tenha aumentado em 7.41% nas turmas de 2016, verifica-se que o alto percentual de erros sugere haver grande dificuldade em assimilar os conceitos básicos de lógica de programação, considerando que se trata de turmas iniciais de algoritmos.

A maioria das questões teve um percentual de respostas incorretas significativamente elevado. A questão 14, por exemplo, teve apenas 48 respostas corretas de 510 submetidas, o que significa que 91% das respostas estavam incorretas. Na Tabela 4.5 é possível verificar que foram as questões das primeiras listas de exercícios as que receberam a maior quantidade de respostas. É possível que as dificuldades iniciais com o raciocínio lógico exigido para resolução das questões apontadas influenciaram para que tantas respostas tenham sido submetidas para questões cujo grau de complexidade não é tão elevado. A Lista 4 aparece em terceiro lugar o que pode indicar que os alunos podem ter enfrentado



Tabela 4.3: Questões ordenadas por Listas de Exercícios

| Questão | Título                         | Lista |
|---------|--------------------------------|-------|
| 01      | Conversão de temperaturas      | 01    |
| 02      | Conversão de velocidade        | 01    |
| 03      | Graus, minutos e segundos      | 01    |
| 04      | Frações                        | 01    |
| 05      | Sistemas de equações           | 01    |
| 06      | Fatores primos                 | 02    |
| 07      | Co-seno de 1 radiano           | 02    |
| 08      | É palíndromo?                  | 02    |
| 09      | É k-alternante?                | 02    |
| 10      | Vigésimo terceiro número primo | 03    |
| 11      | Multiplicação                  | 03    |
| 12      | MDC                            | 03    |
| 13      | Quadrado mágico                | 03    |
| 14      | Números repetidos              | 04    |
| 15      | União                          | 04    |
| 16      | Triângulo de Pascal            | 04    |
| 17      | Aniversário                    | 04    |
| 18      | Intercala                      | 05    |
| 19      | Ocorrências                    | 05    |
| 20      | Gera imagem                    | 05    |
| 21      | Matriz de Vandermonde          | 05    |
| 22      | Deslocamento de imagem         | 06    |
| 23      | Sinuca                         | 07    |

algumas dificuldades nas questões dessa lista de exercícios, possivelmente por causa da inclusão de *arrays*.

Com relação aos tipos de erros, foram registradas 3.771 respostas (41,65%) classificadas automaticamente como **erro de saída**, que são saídas não esperadas, uma vez que a FARMA-ALG exige que a saída do código do aluno seja idêntica à fornecida nos casos de testes. Em seguida, classificadas como **erro de compilação**, foram registradas 2.060

Tabela 4.4: Respostas por questões

| Questão | Lista | Total | 2015     |        |            |       | 2016     |        |            |       |
|---------|-------|-------|----------|--------|------------|-------|----------|--------|------------|-------|
|         |       |       | Corretas | %      | Incorretas | %     | Corretas | %      | Incorretas | %     |
| 01      | 01    | 1.083 | 104      | 17,22  | 500        | 82,78 | 103      | 21,25  | 376        | 78,50 |
| 02      | 01    | 612   | 99       | 32,35  | 207        | 67,65 | 98       | 32,03  | 208        | 67,97 |
| 03      | 01    | 708   | 94       | 25,90  | 269        | 74,10 | 93       | 26,96  | 252        | 73,04 |
| 04      | 01    | 534   | 89       | 35,32  | 163        | 64,68 | 91       | 32,27  | 191        | 67,73 |
| 05      | 01    | 503   | 87       | 29,49  | 208        | 70,51 | 96       | 46,15  | 112        | 53,85 |
| 06      | 02    | 492   | 79       | 23,94  | 251        | 76,06 | 80       | 49,38  | 82         | 80,56 |
| 07      | 02    | 688   | 65       | 18,11  | 294        | 81,89 | 72       | 21,88  | 257        | 78,12 |
| 08      | 02    | 503   | 70       | 29,29  | 169        | 70,71 | 72       | 27,27  | 192        | 72,73 |
| 09      | 02    | 404   | 54       | 23,38  | 177        | 76,62 | 65       | 37,57  | 108        | 62,43 |
| 10      | 03    | 354   | 73       | 34,60  | 138        | 65,40 | 73       | 51,05  | 70         | 48,95 |
| 11      | 03    | 298   | 62       | 31,16  | 137        | 68,84 | 53       | 53,54  | 46         | 46,46 |
| 12      | 03    | 159   | 48       | 55,17  | 39         | 44,83 | 48       | 66,67  | 24         | 33,33 |
| 13      | 03    | 316   | 45       | 26,79  | 123        | 73,21 | 42       | 28,38  | 106        | 71,62 |
| 14      | 04    | 510   | 34       | 11,15  | 271        | 88,85 | 14       | 6,83   | 191        | 93,17 |
| 15      | 04    | 301   | 39       | 21,31  | 144        | 78,69 | 44       | 37,29  | 74         | 62,71 |
| 16      | 04    | 209   | 32       | 33,68  | 63         | 66,32 | 43       | 37,72  | 71         | 62,28 |
| 17      | 04    | 413   | 33       | 14,67  | 192        | 85,33 | 32       | 17,02  | 156        | 82,98 |
| 18      | 05    | 193   | 31       | 36,05  | 55         | 63,95 | 49       | 45,79  | 58         | 54,21 |
| 19      | 05    | 196   | 25       | 25,51  | 73         | 74,49 | 45       | 45,92  | 53         | 54,08 |
| 20      | 05    | 70    | 30       | 100,00 | 0          | 00,00 | 39       | 97,50  | 1          | 2,50  |
| 21      | 05    | 132   | 27       | 34,62  | 51         | 65,38 | 35       | 64,81  | 19         | 35,19 |
| 22      | 06    | 189   | 25       | 21,37  | 92         | 78,63 | 27       | 37,50  | 45         | 62,50 |
| 23      | 07    | 22    | 4        | 66,67  | 2          | 33,33 | 16       | 100,00 | 0          | 00,00 |

Tabela 4.5: Respostas por listas de exercícios

| Lista | Total | %     | 2015      |       | 2016      |       |
|-------|-------|-------|-----------|-------|-----------|-------|
|       |       |       | Respostas | %     | Respostas | %     |
| 01    | 3.441 | 38,72 | 1.821     | 37,52 | 1.620     | 40,17 |
| 02    | 2.085 | 23,46 | 1.157     | 23,84 | 928       | 23,01 |
| 03    | 1.117 | 12,57 | 665       | 13,49 | 462       | 11,46 |
| 04    | 1.443 | 16,24 | 807       | 16,63 | 636       | 15,77 |
| 05    | 591   | 6,65  | 292       | 6,02  | 299       | 7,41  |
| 06    | 188   | 2,12  | 116       | 2,39  | 72        | 1,79  |
| 07    | 22    | 0,25  | 6         | 0,12  | 16        | 0,40  |

respostas (22,75%). Erros de compilação normalmente estão associados a problemas de sintaxe o que denota falta de domínio ou de atenção. A grande quantidade de erros de saída e de compilação pode ser o indício de falta de atenção dos alunos aos enunciados dos problemas de domínio da linguagem de programação utilizada.

Com relação à interação entre alunos e professores, foram registradas apenas 82 respostas (1,93%) que receberam comentários do professor, fato que pode indicar falta de tempo do docente para que fosse possível acompanhar e enviar comentários para todas os alunos que cometeram equívocos em suas respostas. Em quatro casos em que os alunos acertaram as respostas, o professor apenas teceu comentários com o intuito de parabenizar os resultados alcançados pelos alunos. Houve três casos com erro de tempo de execução e outros três casos com erro de saída, que não comprometeram a lógica do algoritmo. Por esses motivos, as respostas foram verificadas e classificadas como corretas pelo professor após o envio de comentários aos alunos alertando-os sobre seus erros.

Ainda sobre a interação, foram registradas apenas 23 mensagens dos professores para grupos de alunos. A maioria das mensagens foi redigida por apenas um professor e envolvem informes gerais sobre a disponibilidade de listas de exercícios. Em alguns casos, o recurso de mensagens foi utilizado para dirimir dúvidas recorrentes de vários alunos, como problemas de erros de compilação e *loop* infinito. Diferentemente dos comentários, que devem ser direcionados para um único aluno, as mensagens permitem que o professor envie uma única informação ou correção para um conjunto de alunos de turmas distintas, ou ainda para uma ou mais turmas.

Dentre as respostas que receberam comentários do professor, foi registrado um número máximo de 37 tentativas para uma única questão. Considerando os casos em que o professor não enviou nenhum comentário, houve casos extremos, como por exemplo, uma questão sobre números repetidos, para a qual um aluno realizou 141 tentativas para tentar resolvê-la. É possível inferir, dada a alta quantidade de tentativas, que o aluno estava tendo dificuldades com conceitos básicos, principalmente de sintaxe, o que foi verificado posteriormente pela observação e comparação de suas respostas.

Ao analisar os erros de compilação, foi verificado que a maior parte deles referem-se

a erros considerados primários. Das 8.887 respostas submetidas, 2.060 (25,9%) foram detectadas com erros de compilação. Desse montante, 1.386 (67,3%) eram erros de *scanner messages*, tais como erros de sintaxe, caracteres ilegais, expressões que excederam a linha e fim inesperado de arquivo. Os erros de *symbol handling*, como identificador não encontrado, identificador duplicado e variáveis não inicializadas, sumarizavam 544 (26,4%). Em seguida, totalizando 500 respostas (24,3%), aparecem os erros de *type checking*, como tipos conflitantes, tipos incompatíveis, conversão ilegal e tipo não declarado.

#### 4.4 Análise crítica das mensagens e comentários

Em um universo de 8.887 respostas submetidas por mais de 229 alunos em sete turmas diferentes, ter ocorrido somente 82 interações sob a forma de comentários é fato que merece atenção. Nesta seção, o teor das mediações efetuadas pelos professores por meio de interações e intervenções proporcionadas pela troca de comentários e mensagens é analisado. Foram analisados critérios quantitativos, como quais questões receberam mais comentários, quantidades de tentativas e quais os tipos de erros mais comuns. A identificação das relações entre as intervenções do professor e o desempenho do aluno, procurou investigar também quais foram os fatores que influenciaram para que o índice de interação tenha sido baixo.

Entre as 8.887 respostas que a FARMA-ALG registrou, algumas delas receberam interações do professor, sob a forma de comentários. Para saber quais respostas deveriam ser selecionadas, foi necessário filtrar somente as que receberam comentários, para que então fosse possível proceder com as análises. De acordo com a Tabela 4.2, foi verificado que em 2015 foram enviados 80 comentários aos alunos e em 2016 apenas dois comentários. A considerável diferença entre as quantidades de comentários de um período para outro imediatamente nos chamou a atenção. O que explicaria o fato de, em 2015, com apenas três turmas termos 80 comentários e em 2016, com quatro turmas, termos apenas dois comentários?

Neste ponto, é necessário contextualizar o planejamento inicial para esta pesquisa e os fatos que aconteceram posteriormente. A primeira proposta objetivava avaliar o impacto

do processo de interação oferecido pela FARMA-ALG, avaliando o quanto todo o processo de ensino-aprendizagem poderia ser beneficiado. Naquele momento, o conjunto de dados disponível para a avaliação eram referentes às três turmas de 2015. Havia, porém, a possibilidade de agregar à pesquisa, os dados de quatro turmas, que iniciaram seus estudos em 2016.

Com dados de diferentes turmas, havia a possibilidade de analisar não somente o impacto da interação, como também e principalmente, o possível **aumento da interação**, entre os períodos e seus possíveis benefícios. Após o término do período de 2016, foi disponibilizado um novo conjunto de dados incrementado aos que estavam disponíveis. Ao acessar a tabela referente aos comentários efetuados, constatou-se que houve um aumento de apenas duas novas entradas registradas.

Apesar da FARMA-ALG oferecer um poderoso mecanismo de interação entre professores e alunos, que funciona com o envio de mensagens e comentários, para que esses recursos sejam plenamente aproveitados, os professores que irão utilizá-la devem estar familiarizados com sua interface e com o fluxo necessário para que seu uso seja efetivo. Entretanto é importante frisar que a FARMA-ALG não possui, até o momento, uma documentação clara e concisa sobre sua metodologia de uso, que aborde os procedimentos necessários para a promoção da interação. Outra situação que chama a atenção, diz respeito à quantidade de professores que enviaram comentários. Havia cinco professores cadastrados na FARMA-ALG e todos estavam habilitados a efetuar interações. Apesar disso, somente um professor utilizou os recursos de interação proporcionados pela ferramenta. Na verdade, não foi o professor que realizou as intervenções, mas o monitor de sua turma, o que reforça a ideia de que os professores não estavam devidamente preparados para sua utilização.

Com o objetivo de compreender o que teria contribuído para um número tão baixo de mensagens, em contato com os professores responsáveis pela utilização da FARMA-ALG em 2016, foi recebida uma explicação do orientador desta pesquisa, via correio eletrônico, no dia 30 de novembro de 2016. A falta de interação entre alunos e professores das turmas novas aconteceu em decorrência de uma série de fatores, detalhados a seguir. Os alunos de doutorado, que ficaram responsáveis em colocar os novos professores em contato direto

com a FARMA-ALG, por diversas circunstâncias, não tiveram as condições necessárias para explicar em detalhes o funcionamento da plataforma, principalmente no que concerne aos aspectos de interação. Como consequência, os professores responsáveis em ministrar a disciplina em que seria utilizada a FARMA-ALG não se sentiram motivados em explorar os recursos de interação, justamente por não tê-los compreendido a contento. Assim, os professores utilizaram a plataforma apenas para cadastrar os exercícios e acompanhar as respostas dos alunos, sem a interação.

Diante dos fatos, foi proposta a elaboração de uma metodologia eficaz de utilização da FARMA-ALG pelos professores, para que mesmo sem o auxílio de outro professor que já tenha domínio da ferramenta, o uso seja orientado do modo correto. A proposta da metodologia, seus detalhes e forma de implementação são detalhadas no Capítulo 5. Na seção seguinte são apresentados os passos metodológicos que foram utilizados para efetuar a análise das mensagens comentários.

#### **4.4.1 Observações para análise crítica dos comentários**

Os seguintes critérios foram considerados para as avaliações dos comentários:

- o aluno, identificado apenas com uma letra;
- a questão para a qual a resposta estava sendo submetida;
- a lista de exercícios da qual a questão faz parte;
- a quantidade de tentativas que o aluno efetuou até ocorrer a primeira intervenção do professor;
- o tempo que o professor demorou até realizar a primeira intervenção;
- a quantidade de tentativas desde a primeira intervenção, até o acerto;
- o tempo desde a primeira intervenção até o acerto.

As análises sobre as respostas corretas, tipo de erros cometidos e a efetividade das interações foram realizadas a partir da observação da relação entre os dados supracitados. Foram considerados quais tipos de situações motivaram os comentários do professor, mesmo para as respostas corretas.

A FARMA-ALG consegue categorizar automaticamente os tipos de erros das respostas erradas através da atribuição de tags. Foram analisados quais os tipos de erros mais comuns em respostas que receberam a intervenção do professor. A partir desses dados, verificou-se como os comentários do professor estão distribuídos entre as questões e os tipos de erros, bem como quais são as questões que os alunos tiveram mais dificuldades.

Também foi realizado um estudo sobre a relação entre certos tipos de erros cometidos em determinadas questões e como a interação do professor influenciou no desempenho dos alunos. Para cada questão que recebeu comentários do professor, verificou-se quantas tentativas o aluno efetuou até que recebesse a intervenção e quanto tempo foi gasto desde a primeira tentativa. A partir da tentativa em que o professor realizou a intervenção, foram verificadas quantas tentativas adicionais foram necessárias e em quanto tempo o aluno conseguiu resolver a questão. Em seguida, foram contabilizadas quantas dessas questões obtiveram a intervenção do professor e foram concluídas com sucesso.

Para avaliar as relações entre a interação e o acerto, foram utilizadas como base as questões em que o aluno não acertou as respostas e recebeu comentários do professor. Primeiramente foi marcada em qual tentativa ocorreu a primeira intervenção e quanto tempo demorou para ela acontecer. Depois, verificou-se quantas tentativas foram necessárias, após a intervenção, para o aluno acertar a resposta e quanto tempo transcorreu. Em seguida, as respostas foram ordenadas de várias formas, procurando relações entre as quantidades de tentativas, o tempo entre elas e como isso poderia influenciar nos acertos.

#### **4.4.2 Resultados da análise crítica dos comentários**

Nesta seção são apresentados os resultados da análise crítica dos comentários trocados entre os professores e seus alunos no decorrer das disciplinas. Na Seção 4.4.2.1 são avaliadas somente as respostas que foram classificadas como corretas. As avaliações efetuadas, considerando os diversos tipos de erros cometidos, seus percentuais e suas relações com os comentários enviados pelo professor, são detalhadas na Seção 4.4.2.2. Na sequência, é avaliada na Seção 4.4.2.3, o quão efetivas foram as intervenções realizadas, verificando sua influência no desempenho dos alunos. Por último, na Seção 4.4.2.4 foram utilizadas

técnicas estatísticas que permitiram verificar as relações entre a interação dispendida e as quantidades de acerto, bem como o tempo e que cada questão demorou para ser resolvida a partir da interação.

#### 4.4.2.1 Respostas corretas

Considerando apenas as 82 respostas que receberam comentários do professor, é possível fazer algumas inferências. Desse total, apenas 10 respostas estavam corretas e a maioria das intervenções do professor refere-se a problemas relacionados a critérios de avaliação da FARMA-ALG, como por exemplo, tempo máximo excedido. É o caso da questão “Quadrado mágico”, em que, apesar de estar correta, a execução para resolução do problema demorou mais tempo do que o previsto pelo caso de teste. Em outro caso da mesma questão, o algoritmo da resposta era de base aleatória, entretanto o aluno teve a sorte em obter a resposta correta na primeira tentativa, de modo que o professor assinalou a resposta como correta, mas alertou para sua aleatoriedade. Também houve casos de uso indevido ou não recomendado de variáveis globais e problemas relacionados à saída não esperada.

Nas situações citadas, o professor considera as respostas como corretas e tece comentários no sentido de orientar o aluno para que ele não incorra mais no mesmo erro. Nessas situações, em que a resposta foi considerada correta pelo professor, foi verificado que suas intervenções tratam de equívocos primários e indicações de boas práticas desejáveis, que influenciam fortemente a sequência do aprendizado do aluno. Houve várias respostas em que a falta de domínio de conceitos básicos de lógica de programação ou da sintaxe da linguagem utilizada se fez perceptível, conforme os exemplos a seguir.

A percepção de que o aluno comete erros primários por falta de domínio de aspectos básicos de lógica se evidencia quando a quantidade de tentativas para responder uma mesma questão e a diferença no código entre as submissões são analisadas. Em uma questão sobre conversão de graus, minutos e segundos, por exemplo, um aluno efetuou 30 tentativas até que conseguisse obter sucesso. Entretanto, ao visualizar o histórico de respostas, foi verificado que até a décima quarta tentativa o aluno enfrentava problemas de



compilação, em sua maioria decorrentes de erros de sintaxe e variáveis não declaradas. A partir da décima quinta tentativa os problemas de compilação foram sanados e as respostas, embora corretas, eram consideradas incorretas pelo sistema devido às diferenças entre a saída do código do aluno e a saída esperada pelos casos de teste cadastrados.

Considerando o breve intervalo entre as sucessivas tentativas, que em alguns casos eram de poucos minutos, nota-se que em várias situações o aluno as efetuou sem ter a clara compreensão do que estava causando o erro. Configura-se portanto uma situação de “tentativa e erro”, sem que se tenha a convicção de que a nova tentativa será efetiva. Vários fatores podem ter contribuído para tal comportamento, como falta de atenção durante a construção do código, falta de conhecimento básico sobre a sintaxe da linguagem utilizada e pouca atenção dedicada à leitura e compreensão do enunciado do problema.

#### 4.4.2.2 Tipos de erros

As respostas que mais receberam comentários do professor foram as que apresentavam algum erro de lógica, que totalizaram 27 respostas, ou 35,53% do total. Em seguida foram as questões que continham diálogos com usuário (não solicitados no enunciado) e erros de saída, com 18 respostas (23,68%). Os problemas com sintaxe aparecem em nove respostas, representando 11,84% e como consequência, alguns deles ocasionaram cinco erros de compilação (6,58%). Alguns alunos submeteram sete respostas (9,21%) cuja solução baseava-se em mecanismos de tentativa e erro, sem nenhuma garantia de que a resposta seria correta.

##### Erros de lógica

A falta de domínio da lógica torna-se clara quando são verificadas no conteúdo dos comentários do professor, frases como *“Não consegui compreender muito bem o seu código. Tente trabalhar da seguinte maneira:”*. Ou ainda *“Estou marcando sua resposta como incorreta pois não consegui compreender sua lógica”*. Em alguns casos, o professor compreendeu a solução proposta, mas mesmo assim alertou o aluno sobre efetividade da solução, que pode ser notada quando o professor comenta *“Entendi o seu código, entretanto*

*“você está utilizando uma forma incorreta de resolver o problema”.*

É notória, entretanto, a preocupação do professor em orientar o aluno da melhor forma possível, concluindo os comentários sempre com uma ideia ou sugestão que contribua para a resolução. Mesmo que o código esteja correto, o professor faz questão de alertar para as boas práticas recomendadas em programação. Essa preocupação é vista, por exemplo, na frase *“Não é necessário fazer vários while’s iguais em sequência. Isso não é uma boa prática. Se há repetição, então ela deve ser colocada em um laço (while, for etc)”*. O uso correto de variáveis também é indicado, como foi verificado quando o professor alerta: *“Parabéns pela resposta. Entretanto, tente não utilizar variáveis globais. Sempre tente passar as informações relevantes para a função através de parâmetros (por valor ou por referência)”*.

### **Erros de saída**

Como a FARMA-ALG espera que a saída do código do aluno seja exatamente igual à saída cadastrada pelo professor nos casos de teste, a quantidade de erros de saída que receberam comentários foi considerável. Foram registradas 18 respostas (23,68%) cujos comentários do professor referenciam problemas com saídas inesperadas. Esses erros foram, em sua maioria, considerados irrelevantes pelo professor, pois se trata de uma particularidade da ferramenta, que ainda não é capaz de classificar uma questão como correta caso haja qualquer diferença da saída do programa elaborado pelo aluno com a saída indicada no enunciado. Nota-se que, para o professor, tal fato não desabona o esforço do aluno em tentar compreender a lógica exigida e resolver o problema.

Merce atenção, entretanto, a quantidade de tentativas que alguns alunos efetuaram até que a resposta fosse considerada correta, embora a lógica da resolução estivesse correta desde o início. Por exemplo, o Código 4.1 refere-se à primeira tentativa de um aluno para resolver um problema de conversão de temperaturas. Embora a lógica do código estivesse correta, a formatação incorreta da saída fez com que a FARMA-ALG não validasse a resposta, pois a saída difere do caso de teste especificado no enunciado. O aluno prosseguiu em várias tentativas, sempre sem sucesso, tendo como retorno variações de erros de saída.

Código 4.1: Conversão de temperaturas (tentativa 1 - erro de saída)

```

1 Program Exercicio_1;
2 var c, n:real;
3 begin
4   read (n);
5   c:= (n*1.8)+32;
6   writeln (c:2:0);
7 end.

```

A intervenção do professor somente ocorreu na tentativa 37, em que ele alerta o aluno sobre a necessidade da saída ser idêntica à especificada no enunciado, orientando-o sobre a sintaxe correta. A resposta só foi considerada correta na tentativa 41, após o aluno ter efetuado as alterações conforme o professor havia recomendado. A linha 6 do Código 4.2 indica que somente a formatação de saída do comando `writeln` foi alterada.

Código 4.2: Conversão de temperaturas (tentativa 41 - erro de saída)

```

1 Program Exercicio_1;
2 var c, n:real;
3 begin
4   read (n);
5   c:= (n*1.8)+32;
6   writeln (c:2:0);
7 end.

```

Um outro exemplo, do mesmo exercício sobre conversão de temperaturas, mostra de forma ainda mais clara como um simples detalhe torna a resposta incorreta. Um aluno efetuou 20 tentativas até que o professor percebesse que o erro estava sendo causado pela ausência de um espaço, conforme segue no comentário a seguir. *“O seu cálculo parece estar correto, tem apenas um detalhe simples faltando. Um espaço entre a resposta e a palavra ‘graus’. Para isso, adicione um espaço no comando writeln (linha 6), exatamente assim: Writeln (f, ‘ graus Fahrenheit’); Abraço!”* Apesar da dica do professor, o aluno só efetuou a alteração necessária na tentativa 38, tendo alternado entre erros de saída e de compilação. Nota-se, portanto, que o recurso de interação que a FARMA-ALG proporciona, permite que o professor possa efetuar suas intervenções com mais facilidade.

No exercício sobre conversão de velocidade, a FARMA-ALG assinalava uma resposta como incorreta devido à falta de delimitadores de casas decimais na saída. Após 16 tentativas sem sucesso, o professor fez a seguinte recomendação ao aluno: *“Adicione :2:2 ao lado da variável do comando write para imprimir apenas 2 casas decimais. Por exemplo: writeln (M:2:2, ‘ ml/h’); Abraço!”*. Tratava-se de um detalhe de formatação da saída, simples de ser resolvido, principalmente considerando que o professor orientou sobre como proceder. Apesar da ajuda, o aluno não deu continuidade e não tentou mais resolver a questão. Por se tratar de um caso isolado, não foi verificada a necessidade em compreender os motivos que o levaram a não responder, mesmo após a intervenção.

O mesmo problema com relação à formatação da saída se repete em outros exercícios, mas sempre contando com a atenção do professor na forma de comentários construtivos. Há a preocupação do professor em, fazendo uso do recurso de retroação ao erro, apontá-lo, sempre com o cuidado de elucidar o que está causando o erro, incentivando o aluno a ele próprio revisar seu código. O seguinte comentário, sobre uma questão de deslocamento de imagem, mostra como o professor se preocupa: *“O primeiro ponto a ser observado é que você está trocando linha por coluna no seu código. Perceba que você lê uma matriz  $3 \times 4$  (exemplo do enunciado) e está imprimindo uma  $4 \times 3$ . Para corrigir isso, você pode trocar apenas a linha 94: De `readln(m,n);` para `readln(n,m);` Entretanto, lembre-se que, nesse caso, você estará chamando linha de  $m$  e coluna de  $n$ . A sua função distorce é outro ponto que está gerando alguns problemas. Eu diria para você mudar sua abordagem: crie duas funções. Uma que realiza apenas UM deslocamento vertical (com um parâmetro que define se o deslocamento é positivo e negativo, e outra que realiza UM deslocamento horizontal) com o mesmo parâmetro. Dessa forma, você apenas executará essas funções o número certo de vezes. Por exemplo, no teste do enunciado, seu programa chamaria a função de deslocamento horizontal 3 vezes, com o parâmetro indicando que é negativo. Depois, chamaria 2 vezes a função de deslocamento vertical, indicando deslocamento positivo. O que acha? Pergunte qualquer dúvida. Abraço!”* A utilização da FARMA-ALG pode ser, portanto, de grande ajuda para que o aluno compreenda o que o levou ao erro de saída, como ele pode corrigi-lo e principalmente, como evitá-lo em exercícios futuros.

## Erros de sintaxe

Os erros de sintaxe figuram em terceiro lugar na lista de questões com erros que receberam algum tipo de intervenção do professor, tendo sido registradas nove respostas (11,84%) para as quais o professor fez comentários sobre os erros de sintaxe cometidos pelos alunos. A maioria desses erros são decorrentes da falta de domínio da sintaxe da linguagem de programação adotada, no caso, Pascal. É importante lembrar, entretanto, que para a disciplina em questão, sempre é disponibilizado um extenso e abrangente guia de referência da linguagem Pascal. A leitura do guia de referência, ainda que não seja obrigatória, é fortemente encorajada pelos professores autores da apostila adotada no curso.

Considerando que se essas orientações fossem seguidas com rigor, provavelmente grande parte dos erros de sintaxe poderiam ser evitados. Fica a impressão de que os alunos desconheciam o referido guia ou não o leram com a devida atenção. Deve-se considerar, todavia, que embora os professores da disciplina tenham informado sobre a importância do guia, é possível que o tenham feito somente no início do período letivo, de modo que, durante o decorrer do semestre, alguns alunos podem ter se esquecido. Salienta-se, principalmente, que a FARMA-ALG não avisa o aluno, quando este inicia sua utilização, sobre quaisquer guias de referência nas linguagens de programação adotadas. Nesse sentido, é sugerido, na Seção 5.3.1, a indicação de leitura dos guias de referência das linguagens de programação utilizadas na FARMA-ALG. Espera-se assim, evitar tais erros nos exercícios, haja vista que eles são consequência do desconhecimento ou falta de domínio da sintaxe da linguagem adotada em cada exercício.

Equívocos simples de sintaxe receberam comentários do professor no sentido de mostrar aos alunos o que está causando o erro e como proceder para dirimi-los. O uso incorreto do operador `mod` é um exemplo e recebeu o seguinte comentário: *“O operador mod precisa receber dois números inteiros. Algo como 2 mod 1”*. Confusões com a separação de parâmetros em chamadas de funções também ocorreram, em que o professor comenta sobre a necessidade do uso de vírgula para esse fim. Até mesmo conceitos básicos, como

o fechamento de comandos com ponto e vírgula, foram negligenciados, como mostra o seguinte comentário: *“Não esqueça o (ponto-e-vírgula) no final de cada comando”*.

### **Erros de compilação**

Como consequência dos problemas de sintaxe, houve várias situações em que o código submetido pelo aluno não compilou corretamente, o que levou o professor a ajudar os alunos com mais dificuldade. Os erros de compilação mais comuns foram os relacionados a tipos incompatíveis, como por exemplo, uma situação em que um `if` espera um resultado booleano, mas retorna um inteiro. Outro caso de incompatibilidade de tipos que gerou erro de compilação é verificado em duas respostas à questão de conversão de graus, minutos e segundos. O erro ocorria porque uma variável estava declarada como inteira, porém o operador devia retornar um número real. Em todos os casos o professor orienta o aluno para que ele identifique e corrija o erro, seja adicionando uma condição lógica que atenda a condição booleana exigida, ou sugerindo o uso do operador correto.

### **Algoritmos de base aleatória**

O professor também teceu comentários sobre um tipo de resposta curiosa, que não estavam necessariamente incorretas, porém continham algoritmos incertos, ineficazes ou aleatórios. Um dos exercícios da Lista 2 consiste em fazer um programa em Pascal capaz de ler apenas um número inteiro positivo do teclado, calcular a decomposição em fatores primos desse número e imprimir os fatores calculados. Em quatro respostas, de diferentes alunos, o professor os alerta para o uso inapropriado e ineficaz do `while`, como pode ser verificado a seguir. *“Estou marcando sua resposta como incorreta pois a utilização de vários while’s para cada número primo não garante que o seu algoritmo sempre apresente a resposta certa. Utilize uma ou duas estruturas de repetição, como while”*. Analisando o código fonte das respostas mencionadas, verifica-se que, de fato, há situações em que o programa não será capaz de apresentar a resposta correta.

Um outra questão, intitulada “Quadrado mágico”, consiste de uma matriz de três linhas por três colunas, de forma que a soma em quaisquer linhas, colunas ou diagonais,

sempre resulte em 15. O enunciado solicita que o aluno crie um programa em Pascal que imprima o primeiro quadrado mágico encontrado. O problema é que alguns alunos utilizaram algoritmos de base aleatória, que podem encontrar a resposta correta, mas que não oferecem nenhuma certeza sobre isso. Para explicar a importância em não utilizar algoritmos aleatórios nas respostas para essa questão, o professor faz duas intervenções.

Na primeira mensagem ele aponta a aleatoriedade da resposta e, ainda que a considere certa, incentiva o aluno para que ele melhore seu algoritmo. *“...como seu algoritmo é de base aleatória, possivelmente ele deu sorte de uma combinação que forma um quadrado mágico ter sido escolhida antes de 1 segundo. Enfim, sua resposta está correta assim. Se quiser não precisa mexer. Mas, se tiver interesse, fique à vontade para tentar otimizá-la”*. Já na segunda mensagem, os resultados não são satisfatórios e o professor explica ao aluno que o motivo é pelo fato da solução ser de base aleatória. *“Sua resposta é 100% randômica. E, por essa razão, pode ser que ela nunca encontre os valores que representam um quadrado mágico. Pode ser, também, que ela encontre de primeira. Mas, infelizmente, a primeira alternativa é que está acontecendo”*. Um exemplo pode ser visto no Código 4.3.

### **Erros de incompatibilidade entre tipos**

Respostas com erros de tipos incompatíveis também receberam a atenção do professor com comentários que orientaram os alunos. Foram cinco respostas (6,58%) cujos resultados estavam incorretos devido a problemas de incompatibilidade de tipos nas atribuições das variáveis. A maioria das respostas com esse tipo de erro foram submetidas na linguagem Pascal, fato que corrobora para a situação discutida anteriormente, sobre a falta de domínio com relação às características básicas da linguagem. Em todos os casos o professor teve o cuidado em explicar a importância em declarar variáveis com tipos compatíveis, para que os alunos, além de corrigir seu código, não incorram mais em erros dessa natureza.

Além de apontar o que está causando o erro, o professor também explica ao aluno os conceitos de tipagem forte, característica de linguagens como Pascal. Foi verificado esse intento no seguinte comentário: *“O Pascal tem o que chamamos de tipagem forte. Ele não aceita que variáveis recebam valores de tipos diferentes dos que foram declarados”*.

Código 4.3: Código que apresenta respostas aleatórias

```

1 program rubyk;
2   var x1, x2, x3, x4, x5, x6, x7, x8, x9: integer;

3   function somas: boolean;
4   begin
5     if (x1 + x2 + x3 = 15) and (x4 + x5 + x6 = 15) and (x7 + x8 + x9 = 15)
6       and (x1 + x4 + x7 = 15) and (x2 + x5 + x8 = 15) and (x3 + x6 + x9 = 15)
7       and (x1 + x5 + x9 = 15) and (x3 + x5 + x7 = 15) then
8       somas:= true
9     else
10      somas:= false;
11   end;

12  begin
13    x1:= 1; x2:= 1; x3:= 1;
14    x4:= 1; x5:= 1; x6:= 1;
15    x7:= 1; x8:= 1; x9:= 1;
16    randomize;
17    repeat
18      x1 := random (10);
19      if x2 = x1 then
20      x2 := random (10);
21      if (x3 = x2) or (x3 = x1) then
22      x3 := random (10);
23      if (x4 = x3) or (x4 = x2) or (x4 = x1) then
24      x4 := random (10);
25      if (x5 = x4) or (x5 = x3) or (x5 = x2) or (x5 = x1) then
26      x5 := random (10);
27      if (x6 = x5) or (x6 = x4) or (x6 = x3) or (x6 = x2) or (x6 = x1) then
28      x6 := random (10);
29      if (x7 = x6) or (x7 = x5) or (x7 = x4) or (x7 = x3) or (x7 = x2) or (x7 =
↵ x1) then
30      x7 := random (10);
31      if (x8 = x7) or (x8 = x6) or (x8 = x5) or (x8 = x4) or (x8 = x3) or (x8 =
↵ x2) or (x8 = x1) then
32      x8 := random (10);
33      if (x9 = x8) or (x9 = x7) or (x9 = x6) or (x9 = x5) or (x9 = x4) or (x9 =
↵ x3) or (x9 = x2) or (x9 = x1) then
34      x9 := random (10);
35      until somas;
36      writeln (x1, ' ', x2, ' ', x3);
37      writeln (x4, ' ', x5, ' ', x6);
38      writeln (x7, ' ', x8, ' ', x9);
39  end;

```



No caso da utilização da linguagem C, o professor também explica com detalhes o que está causando o erro, conforme o comentário a seguir: *“Nesse exercício, não utilize variáveis do tipo real. É por elas que a respostas do último caso de teste está dando errado (no arredondamento do C, quando se converte float para inteiro, perdem-se alguns dados)”*. Complementando, ele orienta o aluno sobre como proceder: *“Para fazer a operação que você está procurando, utilize o operador % (porcentagem). Esse operador retorna o resto de uma divisão entre inteiros. Faça alguns testes e você entenderá”*.

De modo geral, embora a quantidade de interações tenha sido reduzida, em todas elas, com o auxílio da recurso de retroação ao estado de erro, de sua mediação e posterior remediação, por meio da interação que a FARMA-ALG promove, o professor procurou orientar o aluno para a correta resolução dos problemas. Contudo, embora os alunos tenham sido corretamente orientados, não há como afirmar se as intervenções surtiram o efeito desejado, no sentido de, efetivamente, auxiliar para que o erro do aluno seja detectado, mediado e corrigido. Torna-se necessária uma análise aprofundada sobre a efetividade que essas interações produziram sobre as repostas errôneas. Na seção seguinte, avalia-se e discute-se o quão efetivas foram as interações realizadas.

#### 4.4.2.3 Observações sobre a efetividade das interações

Do total de 82 respostas às quais o professor fez alguma intervenção, 71 estavam incorretas e 10 estavam corretas. Em 46 delas (64,78%) os alunos obtiveram avanços, conseguindo identificar e corrigir os erros. Em contrapartida, 25 respostas (35,22%), continuaram incorretas, mesmo com a intervenção do professor. Embora o número de questões que não foram concluídas seja alto – mesmo com o auxílio professor – é perceptível a melhora do desempenho dos alunos que obtiveram ajuda. Das 25 respostas para as quais os alunos não continuaram o desenvolvimento da resolução, a interação do professor contribuiu para que quase dois terços das respostas que estavam incorretas fossem corrigidas.

Considerando as questões cujas respostas receberam comentários mas continuaram incorretas, foi observado que os erros mais comuns estavam relacionados a problemas de

lógica e de saída. Esses tipos de erros continuaram sendo os mais recorrentes, conforme foi visto no início da Seção 4.4.2.2. Observa-se, portanto, que apesar da ajuda do professor ter auxiliado os alunos para que respostas incorretas fossem corrigidas, as dificuldades deles continuam sendo as mesmas, com ou sem a intervenção.

Das 25 respostas incorretas que permaneceram sem novas tentativas, 10 eram relacionadas a problemas na estrutura lógica do código. Para a maioria dos casos, o professor fez comentários no sentido de orientar os alunos para que a lógica do algoritmo adotado fosse corrigida para atender a necessidade do exercício proposto. Todavia, houve casos em que a estrutura proposta parecia bem confusa até mesmo para o professor, como segue no seguinte comentário: *“Em que momento você utiliza a função l ? Aliás, melhore os nomes dessas funções. Caso contrário fica difícil entender o código. ;) Abraço!”*. Ou ainda, em comentários como *“Não entendi essa sua resposta. Precisa de ajuda? Abraço!”* ou também *“Sua resposta para esta questão está ficando um pouco confusa”*.

Outro tipo de erro em respostas incorretas que não tiveram novas tentativas após a intervenção do professor, foram aqueles relacionados a saídas não esperadas. Foram registradas seis respostas com esse tipo de erro, em que a saída do código do aluno não corresponde com a saída esperada pelos casos de teste cadastrados na FARMA-ALG. Trata-se de um problema recorrente sobre a falta de atenção dos alunos com relação a necessidade de formatar a saída corretamente. No decorrer da pesquisa, esse tipo de erro passa a ocorrer com maior frequência nos exercícios da primeira lista. Conforme o professor percebia esses erros e fazia intervenções no sentido de orientar sobre a formatação correta, eles passaram a aparecer cada vez menos nas questões seguintes.

A quantidade de erros de saída nos primeiros exercícios merece atenção, pois havia o entendimento de que os alunos estavam cientes sobre como deviam proceder para que as saídas fossem idênticas às dos casos de teste. Observa-se que em cada turma esses erros se repetiam nas questões das primeiras listas, fazendo com que o professor tivesse que enviar comentários idênticos sobre esse problema para alunos de todas as turmas. Tal procedimento, certamente, requer o tempo e atenção tanto do professor – que precisa ler as respostas, formular e enviar os comentários – quanto dos alunos, que precisam reenviar

suas respostas com as correções de formatação de saída adequadas. Considerando que o aluno tenha sido previamente informado sobre a estrita necessidade de submeter seu código de modo que sua saída tenha que ser idêntica às dos casos de teste, o que estaria acontecendo para que tais erros continuassem a ocorrer?

A primeira hipótese seria a falta de atenção dos alunos, pois em alguns casos, mesmo após a intervenção do professor, haviam respostas que continuaram com erros de saída. No entanto, em nova consulta ao banco de dados com os comentários do professor, foi constatado que em cada mensagem ele sempre enfatiza a necessidade da saída ser a mesma do caso de teste, como verificado a seguir: *“No Farma-Alg é importante que a saída siga exatamente o padrão apresentado no enunciado. Sem nenhuma diferença, por menor que seja”*. Se o aluno já sabe como deve proceder, por qual motivo o professor enfatiza tanto que a saída deve seguir esse padrão? Ou será que, exatamente pelo fato dessa obrigatoriedade não ter ficado explícita, ele viu a necessidade de explicar através dos comentários?

Ao iniciar a resolução de uma questão, a primeira ação que o aluno deve realizar é ler atentamente o enunciado do problema proposto. É natural acreditar no pressuposto de que, se há a necessidade da saída do código do aluno ser impreterivelmente idêntica à saída do caso de teste, ela deve estar explicitada no enunciado de cada questão. Para averiguar sobre essa condição, foi verificado se ela constava em alguma parte nos textos dos enunciados das questões cadastradas nos Objetos de Aprendizagem. Para tanto, as páginas da FARMA-ALG referentes aos OAs foram acessadas e os enunciados de cada exercício proposto foram verificadas um a um.

Em todos os enunciados havia exemplos de como deviam ser a entrada e qual seria a saída esperada para aquele exercício. Não havia, entretanto, nada que indicasse ao aluno que a saída do seu programa deveria ser idêntica às saídas dos casos de teste. Para além da obrigatoriedade, também não havia nenhum alerta explícito, que advertisse o aluno que, caso a saída da sua resposta fosse diferente, ainda que minimamente, a FARMA-ALG consideraria a resposta como incorreta. Havia, portanto, o indício de que os recorrentes erros de saída cometidos pelos alunos poderiam ser decorrentes do seu desconhecimento

sobre essa exigência da ferramenta.

Com o objetivo de compreender por quais motivos os enunciados não continham nenhuma referência a respeito do formato de saída exigido, o professor que cadastrou os exercícios na ferramenta e criou os enunciados foi contatado. Em resposta, o professor informa que no momento em que os enunciados foram criados, esses detalhes passaram despercebidos, mas que os alunos foram informados através de um e-mail. Neste e-mail, os alunos recebem instruções sobre itens básicos, como por exemplo, efetuar o cadastro, a matrícula e como acessar sua Turma. Além disso, há instruções sobre como as respostas devem ser submetidas na FARMA-ALG, para evitar que elas sejam interpretadas como incorretas.

No trecho a seguir, o professor informa aos alunos a necessidade de se seguir o exemplo de entrada e saída: *“Vale ressaltar que, ao submeter respostas ao Farma-Alg, é necessário seguir exatamente o exemplo de entrada e saída dado no enunciado da questão no FARMA-ALG”*. Ele também chama a atenção sobre não haver necessidade de diálogos: *“Não é necessário nenhum tipo de diálogo com o usuário como ‘Entre com os dados’ ou ‘Digite a temperatura’”*. Finalmente, alerta para a possibilidade de erro, caso essas regras não forem seguidas: *“Se alguma saída além da especificada no enunciado for encontrada o sistema irá considerar a resposta incorreta, por mais que ela apresente o resultado correto”*.

Considerando que o e-mail foi enviado uma única vez – apenas no início do período letivo – e que não havia instruções mais detalhadas nos enunciados, é compreensível a quantidade de erros de saída cometidas pelos alunos. Conforme os alunos iam avançando nas próximas questões, contando inclusive com a ajuda do professor, esses erros começaram a diminuir. O tempo que os alunos levaram para elaborar respostas de acordo com as saídas exigidas e a redução de erros após um certo tempo de uso, também foram confirmados pelo professor, conforme resposta via e-mail: *“Os alunos precisam de tempo e apresentam dificuldades na adaptação de seus programas para saídas pré-determinadas. Entretanto, após uso da ferramenta, esses problemas tendem a diminuir bastante”*. É possível que, se as orientações necessárias a respeito das saídas dos códigos tivessem sido informadas com detalhes, nos enunciados, a quantidade de erros desse tipo fosse consideravelmente menor.

#### 4.4.2.4 Relações entre a interação e o acerto

Como a maioria dos alunos acertou as questões que estavam com dificuldades após a interação do professor, há a dúvida se o tempo transcorrido da primeira tentativa até a intervenção teria influenciado de alguma forma. O intento seria procurar possíveis padrões de tempo até a interação, que pudessem evidenciar a possível eficácia do processo de intervenção dentro de um certo intervalo de tempo e tentativas. Considerando em qual tentativa ocorreu a primeira intervenção, seria possível que as quantidades de tentativas até o acerto seriam reduzidas?

Em princípio, havia a expectativa que, a partir do momento em que o aluno tenta responder uma questão e comete um erro, quanto mais cedo ocorresse a intervenção do monitor, por conseguinte menos tentativas seriam necessárias para que ele corrigisse o código e acertasse a resposta. Os dados de tempo da intervenção até o acerto foram ordenados de forma crescente, o que possibilitou a verificação de que, na maioria das vezes, a primeira intervenção ocorreu em poucos minutos, conforme a Tabela 4.6. Apenas em duas situações, no caso do aluno F, o monitor demorou um pouco mais para interagir, nas questões 01 e 15. Tal fato não corrobora, todavia, para que seja correto afirmar que quanto antes ocorrer a interação, mais cedo será a possibilidade do aluno corrigir e acertar a questão.

Como exemplo, o aluno F foi respondido em apenas 17 minutos, porém demorou 54 dias para submeter a resposta correta para a questão 13. De modo similar, o aluno M só obteve sucesso na questão 04 após quase 30 dias desde que o monitor interagiu com ele, somente 2 minutos após a primeira tentativa. Todavia, em todas as respostas para as quais o professor demorou para enviar um comentário, o aluno também demorou consideravelmente para acertar. Ou seja, se por um lado, interagir rapidamente não pode ser considerado como um diferencial para que o aluno acerte mais cedo, por outro lado, a interação tardia pode desmotivá-lo, contribuindo para que ele não se dedique às soluções com mais afinco. Como a avaliação dos resultados desta seção é baseada em análise empírica, é recomendável, para trabalhos futuros, análises com dados de mais turmas para que se possa fazer asserções mais seguras sobre as relações entre a interação e o acerto.

Tabela 4.6: Tempo e tentativas após intervenções

| Aluno | Questão | Lista | Tentativa<br>da 1 <sup>a</sup><br>intervenção | Tempo<br>até a 1 <sup>a</sup><br>intervenção | Tentativa<br>em que<br>acertou | Tentativas<br>da<br>intervenção<br>até o acerto | Tempo da<br>intervenção até o<br>acerto |
|-------|---------|-------|---|--|--------------------------------|---|---|
| P     | 02      | 01    | 3   | 3m   | 4                              | 1   | 1m                                      |
| E     | 01      | 01    | 1   | 0m   | 3                              | 2   | 7m                                      |
| U     | 10      | 03    | 6   | 33m  | 7                              | 1   | 15m                                     |
| S     | 01      | 01    | 1   | 0  | 2                              | 1   | 29m                                     |
| Q     | 03      | 01    | 14  | 32m  | 30                             | 16  | 41m                                     |
| B     | 04      | 01    | 3   | 6m   | 5                              | 2   | 1h 13m                                  |
| J     | 12      | 03    | 1   | 0  | 2                              | 1   | 1h 38m                                  |
| F     | 01      | 01    | 28  | 1d 18h                                       | 38                             | 10  | 2h 20m                                  |
| H     | 22      | 06    | 1   | 0  | 7                              | 6   | 2h 22m                                  |
| V     | 01      | 01    | 10  | 34m  | 11                             | 1   | 4h 5m                                   |
| K     | 03      | 01    | 1   | 0  | 2                              | 1   | 5h 49m                                  |
| R     | 07      | 02    | 34  | 15h 49m                                      | 44                             | 10  | 7h 37m                                  |
| G     | 22      | 06    | 3   | 6h 25m                                       | 5                              | 2   | 14h 3m                                  |
| S     | 03      | 01    | 5   | 8h 7m  | 14                             | 9   | 16h                                     |
| S     | 02      | 01    | 2   | 0  | 4                              | 2   | 16h 29m                                 |
| Q     | 01      | 01    | 37  | 3h 13m                                       | 41                             | 4   | 17h 46m                                 |
| O     | 06      | 02    | 2   | 59m  | 3                              | 1   | 19h 33m                                 |
| L     | 01      | 01    | 8   | 24m  | 12                             | 4   | 20h 25m                                 |
| Z     | 17      | 04    | 12  | 3d 6h  | 15                             | 3   | 22h 17m                                 |
| V     | 04      | 01    | 2   | 1m   | 3                              | 1   | 23h 36m                                 |
| S     | 03      | 01    | 3   | 0  | 14                             | 11  | 24h 8m                                  |
| R     | 10      | 03    | 5   | 4m   | 22                             | 17  | 1d 4h                                   |
| F     | 01      | 01    | 20  | 14h 55m                                      | 38                             | 18  | 1d 5h                                   |
| X     | 06      | 02    | 2   | 2m   | 3                              | 1   | 1d 7h                                   |
| U     | 02      | 01    | 6   | 3m   | 12                             | 6   | 1d 12h                                  |
| U     | 01      | 01    | 12  | 10h 37m                                      | 15                             | 3   | 1d 12h                                  |
| Y     | 01      | 01    | 7   | 36m  | 10                             | 3   | 1d 23h                                  |
| D     | 14      | 04    | 5   | 01h 6m                                       | 15                             | 10  | 2d 1h                                   |
| J     | 04      | 01    | 3   | 1h 26m                                       | 4                              | 1   | 2d 4h                                   |
| A     | 13      | 03    | 2   | 0  | 3                              | 1   | 2d 7h                                   |
| F     | 08      | 02    | 13  | 7m   | 18                             | 5   | 2d 20h                                  |
| D     | 08      | 02    | 25  | 1h 27h                                       | 25                             | 0   | 3d 5h                                   |
| I     | 01      | 01    | 3   | 6m   | 7                              | 4   | 3d 22h                                  |
| F     | 16      | 04    | 17  | 1h 23m                                       | 26                             | 9   | 4d                                      |

*Continua na próxima página*

Tabela 4.6 – *Continuação da página anterior*

| Aluno | Questão | Lista | Tentativa<br>da 1 <sup>a</sup><br>intervenção | Tempo<br>até a 1 <sup>a</sup><br>intervenção | Tentativa<br>em que<br>acertou | Tentativas<br>da<br>Intervenção<br>até o acerto | Tempo da<br>intervenção até o<br>acerto |
|-------|---------|-------|---|--|--------------------------------|---|---|
| F     | 04      | 01    | 5   | 46m  | 19                             | 14  | 4d                                      |
| X     | 07      | 02    | 11  | 49m  | 12                             | 1   | 6d                                      |
| T     | 10      | 03    | 7   | 3m   | 9                              | 2   | 6d 8h                                   |
| N     | 01      | 01    | 12  | 2h 8m  | 18                             | 6   | 6d 15h                                  |
| X     | 11      | 03    | 2   | 0  | 3                              | 1   | 6d 22h                                  |
| F     | 03      | 01    | 9   | 27m  | 30                             | 21  | 8d 4h                                   |
| R     | 18      | 05    | 28  | 2h 52m                                       | 29                             | 1   | 8d 14h                                  |
| M     | 15      | 04    | 2   | 2m   | 3                              | 1   | 29d 20h                                 |
| R     | 15      | 04    | 14  | 28h 1m                                       | 17                             | 3   | 36d 4h                                  |
| F     | 15      | 04    | 25  | 6d 1h  | 26                             | 1   | 36d 19h                                 |
| F     | 13      | 03    | 7   | 17m  | 16                             | 9   | 54d 18h                                 |
| C     | 03      | 01    | 6   | 20h 1m                                       | 10                             | 4   | 88d 6h                                  |

## 4.5 Análise estatística dos comentários

É incorreto considerar, entretanto, somente a análise empírica dos dados tabulados para que sejam feitas asserções. É possível que exista uma relação entre a ocorrência da primeira intervenção e o acerto do aluno, em termos de quantidade de tentativas e tempo decorrido. Para que verificar tal condição, foram utilizadas técnicas de testes de hipóteses com duas amostras para comparar as quantidades de tentativas que ocorreram até a primeira intervenção, com a quantidade de tentativas desde a intervenção até o acerto.

### 4.5.1 Metodologia da análise estatística dos comentários

A seção anterior teve como objetivo a comparação, de modo empírico e crítico, das mensagens e comentários que houve entre alunos e professores. Para compreender melhor a questão sobre o quanto a interação pode contribuir para a melhoria do desempenho do aluno, os dados foram submetidos a análises estatísticas descritiva e inferencial. Os dados de interesse foram dispostos em dois vetores de igual dimensão, tendo como base as

questões em que houve algum tipo de interação entre professor e aluno e cuja resposta foi correta. Com base nesse critério, das 82 respostas com interação, restaram apenas 46 respostas corretas, que receberam algum tipo de interação. Os dados analisados são os mesmos dispostos na Tabela 4.6.

Os dados de interesse para as análises estatísticas foram:

- a quantidade de tentativas até que ocorresse a primeira intervenção (tent\_1\_int);
- a quantidade de tentativas desde a primeira intervenção, até o acerto (tent\_int\_ok);
- o tempo que o professor demorou para realizar a primeira intervenção (temp\_1\_int);
- o tempo, desde a primeira intervenção, até o acerto (temp\_int\_ok).

Como citado anteriormente, foram efetuadas dois tipos de análise:

- **análise descritiva:** na análise descritiva, para o conjuntos de dados supracitado, foram comparadas as médias, os desvios padrão e os coeficientes de variação. Os resultados das comparações são apresentados em gráficos Boxplot.
- **análise inferencial:** na análise inferencial, os mesmos dados foram submetidos a testes estatísticos para comparação de médias para validar ou não, os resultados observados na análise descritiva.

Devido às características amostrais tenderem ao aspecto inferencial das análises, a metodologia adotada foi organizada segundo o fluxograma concebido por Ramos (2017), adaptado pelo autor e disposto na Figura 4.3. A concepção da metodologia e aplicação das técnicas descritivas e inferenciais, utilizadas nas análises desta pesquisa, teve a supervisão e validação dos resultados pelo Curso de Matemática<sup>7</sup> da Universidade do Estado de Mato Grosso (UNEMAT), sob supervisão do Prof. Me. João Gabriel Ribeiro. Cada passo do fluxograma é detalhado a seguir.

#### 4.5.1.1 Teste de normalidade (Teste Shapiro-Wilk)

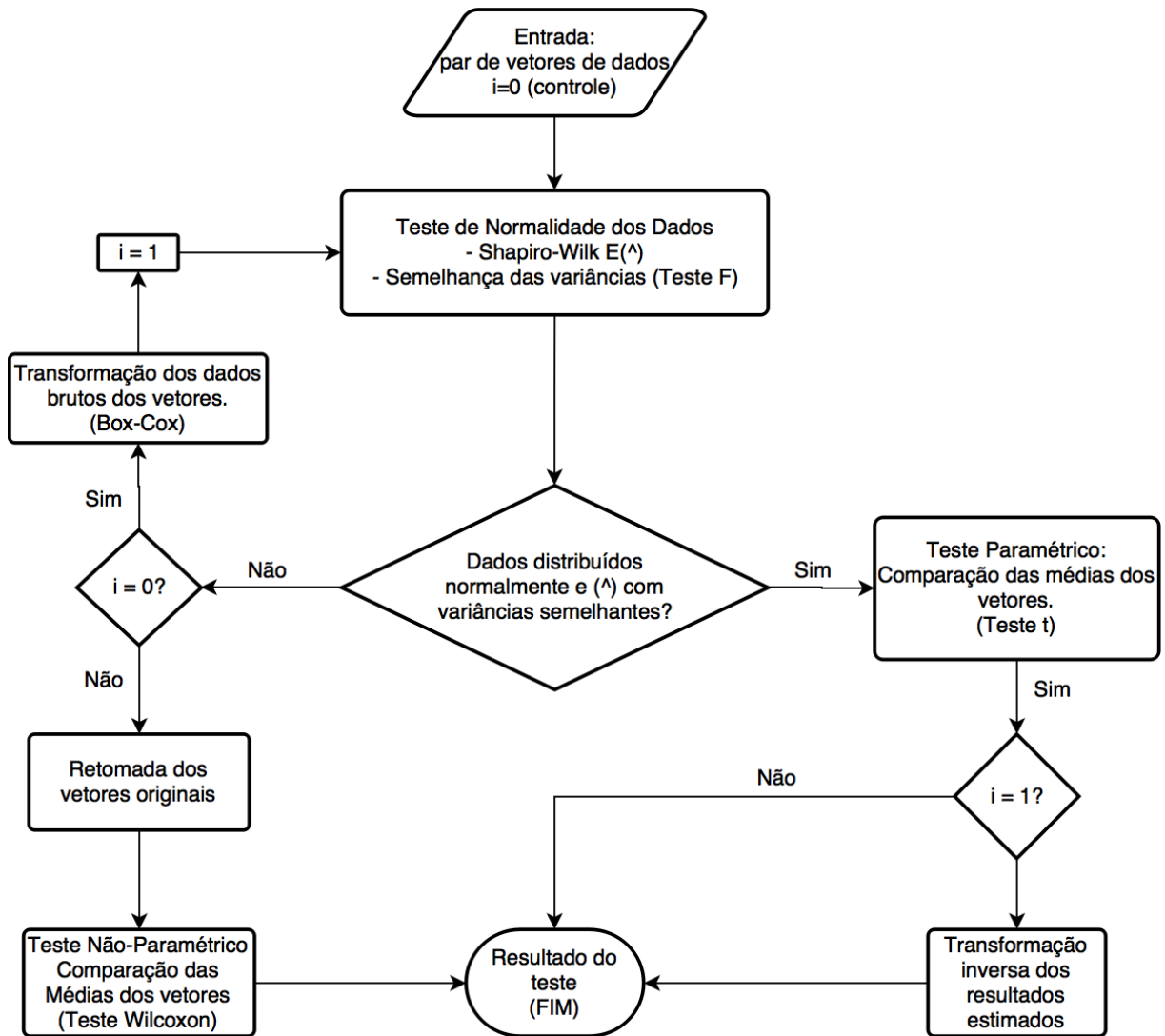
O primeiro passo consiste em investigar se dados que serão comparados possuem uma distribuição considerada normal e é realizado por meio do Teste de Shapiro-Wilk. O

---

<sup>7</sup> <<http://sinop.unemat.br/site/faculdades/facet/curso-de-matematica/>>



Figura 4.3: Fluxograma de análise estatística inferencial



Fonte: (RAMOS, 2017). Adaptado pelo autor.

procedimento consiste em considerar em nível de significância  $\alpha$  (em geral, 5%) e ocorre da forma descrita a seguir. Neste método, duas possíveis hipóteses são consideradas:

1.  $H_0$ : Amostras têm origem em uma população normal  $N \sim (\mu, \sigma)$ ;
2.  $H_1$ : Amostras não têm origem em uma população normal.

O cálculo de estatística do teste é dado por:

$$W = \frac{b^2}{\sum_{i=1}^n (x_{(i)} - \bar{x})^2} \quad (4.1)$$

de modo que  $x_{(i)}$  são os valores ordenados das amostras (em que  $x_{(1)}$  é o menor deles)

e  $\bar{x}$  é a média do vetor. A constante  $b$  é determinada por meio de:

$$b = \begin{cases} \sum_{i=1}^{n/2} a_{(n-i+1)}(x_{n-i+1} - x_{(i)}) & \text{se } n \text{ é par} \\ \sum_{i=1}^{(n+1)/2} a_{(n-i+1)}(x_{n-i+1} - x_{(i)}) & \text{se } n \text{ é ímpar} \end{cases} \quad (4.2)$$

em que  $a_{(n-i+1)}$  é uma constante gerada pela média, variância e covariância da estatística de ordem de uma amostra de tamanho  $n$  de uma distribuição Normal.

Para determinar se há a normalidade ou não, rejeita-se  $H_0$  ao nível de significância  $\alpha$  se  $W > W_\alpha$ , sabendo que os valores críticos da estatística  $W$  de Shapiro-Wilk são dados tabelados.

#### 4.5.1.2 Teste de comparação entre variâncias (Teste F)

Para que seja possível efetuar a comparação das médias por meio de testes paramétricos, supondo que as condições de normalidade estão satisfeitas, é recomendável utilizar o Teste F. Para isso, é necessário que a variância entre os pares de vetores comparados sejam semelhantes. Considerando um nível de significância  $\alpha$  (frequentemente, de 5%), o procedimento ocorre da seguinte forma, tomando as hipóteses que estão em análise:

Comparando a quantidade de tentativas até a primeira interação com a quantidade de tentativas desde a primeira interação até o acerto:

$$H_0 : \sigma_{tent-int}^2 = \sigma_{tent-ok}^2$$

$$H_1 : \sigma_{tent-int}^2 \neq \sigma_{tent-ok}^2$$

Comparando o tempo gasto até a primeira interação com o tempo desde a primeira interação até o acerto:

$$H_0 : \sigma_{temp-int}^2 = \sigma_{temp-ok}^2$$

$$H_1 : \sigma_{temp-int}^2 \neq \sigma_{temp-ok}^2$$

Efetua-se então o cálculo da estatística do teste para tomada de decisão:

$$F = \frac{\sigma_{tent-int}^2}{\sigma_{tent-ok}^2} \quad (4.3)$$

Considerando o teste bilateral, determina-se os pontos críticos  $F_{\alpha/2}$  e  $F_{1-\alpha/2}$  da distribuição  $F$  com  $n_1 - 1$  graus de liberdade no numerador e  $n_2 - 1$  graus de liberdade no denominador usando a tabela de distribuição Fisher-Snedecor, de modo que  $\mathbb{P}[F < F_{\alpha/2}] = \mathbb{P}[F > F_{1-\alpha/2}] = \alpha/2$ .

#### 4.5.1.3 Teste paramétrico de comparação das médias (Teste t)

Caso os pressupostos de normalidade dos dados e de semelhanças das variâncias sejam atendidos, utiliza-se o Teste t para a comparação das médias. É utilizada sua forma bilateral com nível de significância  $\alpha$  (frequentemente como 5%). As hipóteses são as estabelecidas como segue:

$$H_0 : \mu_{temp-int} = \mu_{temp-ok}$$

$$H_1 : \mu_{temp-int} \neq \mu_{temp-ok}$$

O cálculo da estatística do teste, para tomada de decisão, é dado por:

$$t = \frac{\mu_{temp-int} - \mu_{temp-ok}}{\sqrt{\frac{\sigma_{temp-int}^2}{n_{temp-int}} - \frac{\sigma_{temp-ok}^2}{n_{temp-ok}}}} \quad (4.4)$$

Considerando o teste bilateral, sobre a a igualdade das variâncias, para a tomada de decisão rejeita-se  $H_0$  ao nível de significância  $\alpha$ , se  $|t| \geq |t_{\alpha/2}|$ . o pontos críticos  $t_{1-\alpha/2}$  e  $t_{\alpha/2}$  da distribuição  $t$  provêm da tabela da distribuição  $t$  Student, de modo que  $P[t \leq t_{\alpha/2}] = P[t \geq t_{1-\alpha/2}] = \alpha/2$ .

#### 4.5.1.4 Transformação dos dados (Box-Cox)

Caso as amostras não atendam aos pressupostos de normalidade e homogeneidade de variâncias, é necessário transformar os dados. A solução mais indicada para o caso é a transformação Box-Cox, em que uma entrada não normalizada possa produzir uma saída normalizada. Considerando  $x_1, \dots, x_n$  como sendo os dados originais, a transformação Box-Cox consiste em encontrar um  $\lambda$  de modo que  $y_1, \dots, y_n$  se aproximem ao máximo de uma distribuição normal. Esta transformação é dada por:

$$y_i(\lambda) = \begin{cases} \ln(x_i) & \text{se } \lambda = 0 \\ \frac{x_i^\lambda - 1}{\lambda} & \text{se } \lambda \neq 0 \end{cases} \quad (4.5)$$

Após a transformação, os testes referentes aos pressupostos são novamente aplicados, utilizando os novos vetores resultantes da transformação Box-Cox. Caso os pressupostos sejam satisfeitos, aplica-se o Teste t novamente e executa-se a transformação inversa com o objetivo de capturar inferências das médias e outros parâmetros.

Se, ainda assim, os pressupostos não sejam atendidos após a transformação, recorre-se às ferramentas de análises estatísticas não-paramétricas. O teste de Wilcoxon é aplicado nos dados das médias originais utilizando-se as mesmas hipóteses. Utiliza-se testes não-paramétricos nesses casos, pois eles não exigem pressupostos sobre a normalidade da distribuição das médias e de seu comportamento de variação.

#### 4.5.1.5 Teste não-paramétrico de comparação das médias (Teste de Wilcoxon)

Considerando a possibilidade das amostras não possuírem distribuição normal nem serem homogêneas em suas variâncias, o Teste t não deve ser utilizado. Para esses casos, utiliza-se um teste não-paramétrico para comparação das médias. O teste não-paramétrico de Wilcoxon é então utilizado, testando as seguintes hipóteses:

$$H_0 : \mu_{temp-int} = \mu_{temp-ok}$$

$$H_1 : \mu_{temp-int} \neq \mu_{temp-ok}$$

Para tomada de decisão, efetua-se o cálculo da estatística do teste, sendo  $T^+$  ou  $T^-$ , com a soma a ambos sempre constante, de acordo com:

$$W_2 = \frac{(n_1 + n_2) \cdot (n_1 + n_2 + 1)}{2} - W_1 \quad (4.6)$$

De modo que  $W_1$  é o menor valor obtido da soma dos postos do vetor  $n_1$  e  $W_2$  a soma dos postos do vetor  $n_2$ .

Para a tomada de decisão sobre o contraste das médias, os seguintes procedimentos são tomados: Rejeita-se  $H_0$  ao nível de significância  $\alpha$  se  $W_2 \leq W_{\alpha/2}$  e  $W_2 \geq W_{1-\alpha/2}$ . Os valores críticos  $W_{\alpha/2}$  e  $W_{1-\alpha/2}$  da estatística são tabelados.

## 4.5.2 Resultados da análise estatística dos comentários

Nesta seção são apresentados os principais resultados oriundos dos testes estatísticos efetuados. Primeiramente, são apresentados resultados dos testes comparativos, com análises descritivas do desempenho dos alunos, considerando a intervenção do professor. Posteriormente são vistos os resultados dos testes de análise inferencial. Todos os testes foram efetuados usando a linguagem R e o software R-Studio. A rotina utilizada pode ser vista no Apêndice 15.

### 4.5.2.1 Quantidade de tentativas

Os dados referentes às quantidades de tentativas foram submetidos aos testes de estatística descritiva para obtenção dos valores de média, desvio padrão e coeficientes de variância. Os dados podem ser vistos na Tabela 4.7.

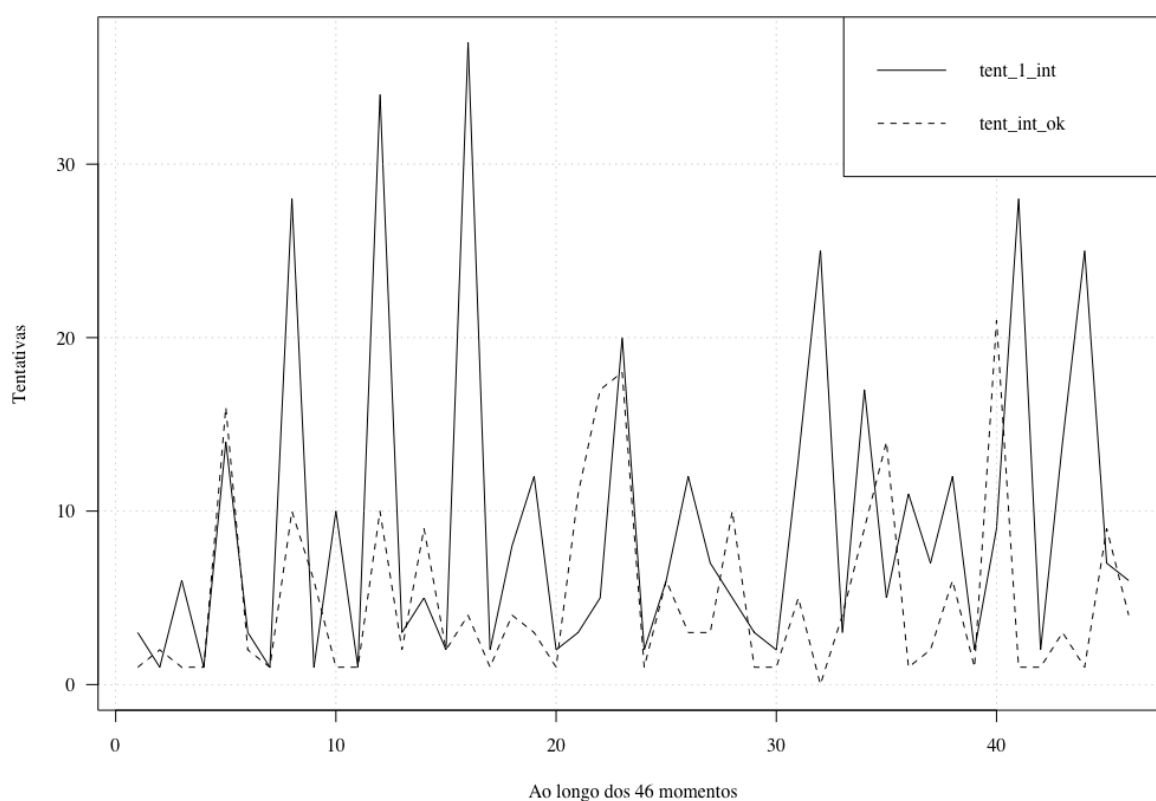
Em geral, a Figura 4.4 mostra que os alunos efetuaram uma quantidade de tentativas consideravelmente maior até que recebessem a primeira intervenção do professor, em comparação à quantidade de tentativas necessárias, após a intervenção, para que o problema fosse solucionado. De fato, considerando as médias da Tabela 4.7, houve uma redução

Tabela 4.7: Média, Desvio Padrão e Coeficiente de Variância das tentativas

| Tentativas                                       | Média | DP   | CV     |
|--|-------|------|--------|
| Até primeira interação (tent_1_int)              | 9,23  | 9,31 | 100,84 |
| Da primeira interação até o acerto (tent_int_ok) | 5,02  | 5,33 | 106,16 |

de 46% na quantidade de tentativas até que o aluno obtivesse sucesso em sua resposta após a primeira interação. A interação do professor contribuiu, portanto, para que o aluno conseguisse dirimir eventuais dúvidas que o impediavam de prosseguir.

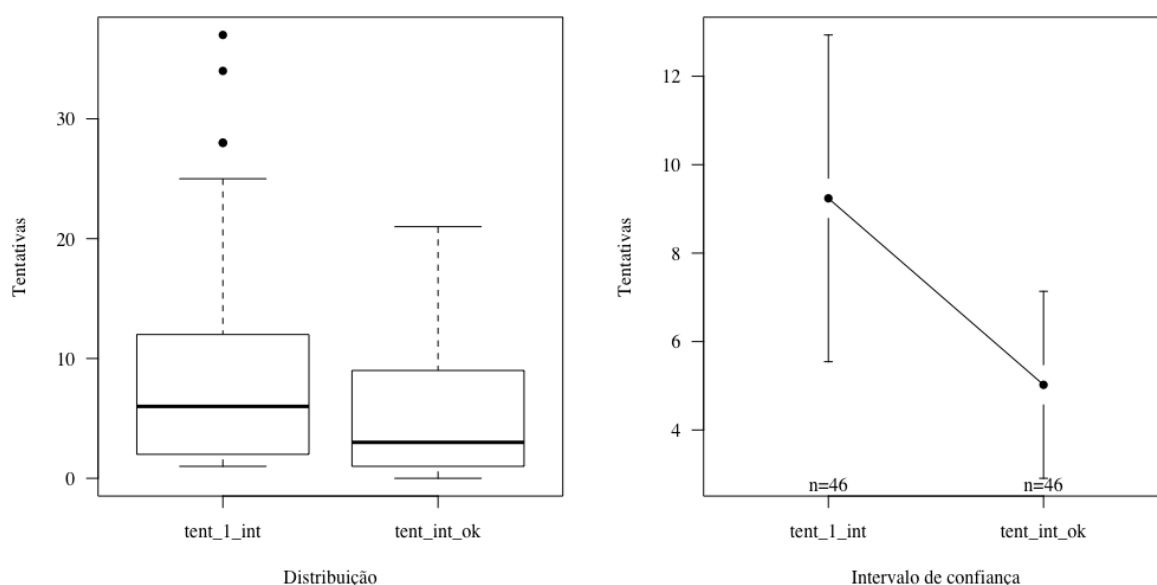
Figura 4.4: Comparativo entre as tentativas executadas até a primeira interação e desde a interação até o acerto



Ao observar o boxplot da Figura 4.5, verifica-se que a maior parte dos dados antes da primeira interação (`tent_1_int`), se concentram em intervalo mais amplo e com valores ligeiramente maiores que os dados relativos às respostas após a interação (`tent_int_ok`). É perceptível, portanto, considerando a menor amplitude da distribuição dos dados do

boxplot à direita, que a quantidade de tentativas sofreu considerável redução após o professor ter feito a primeira intervenção. Observa-se também, que antes da primeira interação, alguns alunos tendem a demonstrar desconhecimento ou falta de domínio do conteúdo. Isso pode ser observado pelos *outliers* superiores no boxplot à esquerda, que indicam um alto número de tentativas sem sucesso.

Figura 4.5: Boxplot e Plotmeans comparando as tentativas executadas até a primeira interação e desde a interação até o acerto



Seguindo o fluxograma disposto na Figura 4.3, foram efetuados testes de normalidade de dados. O primeiro teste realizado foi o de Shapiro-Wilk, com o intuito de verificar se os dados apresentam uma distribuição considerada normal. Para indicar normalidade, é necessário que o *p-value* seja maior que o nível de significância  $\alpha$ , no caso, 5%. Para os dados da quantidade de tentativas até que ocorresse a primeira interação, o teste apresentou  $W = 0,80$  e  $p\text{-value} = 2,272 \times 10^{-6}$ . Já os dados sobre a quantidade de tentativas desde a primeira interação até o acerto retornaram  $W = 0,78$  e  $p\text{-value} = 9,88 \times 10^{-7}$ .

Os *p-value* são extremamente baixos, indicando claramente que nenhum dos conjuntos de dados apresentou uma distribuição normal. Além da verificação da normalidade da distribuição, se faz necessária a confirmação da homogeneidade das variâncias por meio do

Teste F. Os resultado do Teste F apontaram  $F = 3,05$  e  $p\text{-value} = 2,787 \times 10^{-4}$ , valores que não confirmam a homogeneidade das variâncias. Como os dados não obedeceram os pressupostos de normalidade e homogeneidade, faz-se necessária, portanto, a transformação dos dados. A transformação dos dados foi efetuada pelo método de Boxcox, com  $\lambda = 0,73$ .

Após a transformação dos dados, os testes de normalidade das distribuições foram retomados. Para a primeira série de dados, relativa à quantidade de tentativas até a primeira interação, os valores obtidos foram  $W = 0,86$  e  $p\text{-value} = 8,42 \times 10^{-5}$ . Para a segunda série de dados, sobre o tempo desde a primeira interação até o acerto, o teste retornou  $W = 0,84$  e  $p\text{-value} = 2,01 \times 10^{-5}$ . Concluimos que mesmo após a transformação dos dados, os testes continuam indicando a não normalidade dos conjuntos avaliados.

Também foi verificada novamente a homogeneidade das variâncias após a transformação. Neste caso, o resultado apresentado foi de  $F = 2,18$  e  $p\text{-value} = 9,87 \times 10^{-3}$ . Assim como foi detectada a não normalidade dos dados no passo anterior, o comportamento da homogeneidade também demonstrou-se fora dos padrões aceitáveis. Nesse sentido, de acordo com o fluxograma, os dados originais de antes da transformação foram retomados para execução do teste não paramétrico.

Como todos os pressupostos de normalidade e homogeneidade não foram atendidos, mesmo após a transformação dos dados, foi necessário utilizar métodos não paramétricos. Assim, efetuou-se a comparação das médias por meio do teste não paramétrico de Wilcoxon. Os resultados obtidos foram  $W = 1407$  e  $p\text{-value} = 6,07 \times 10^{-3}$ , o que indica que há diferenças expressivas entre as médias.

#### 4.5.2.2 Tempo entre as tentativas

Após a análise da quantidade de tentativas, foram efetuadas análises descritivas sobre os tempos entre elas, obtendo os valores da média, desvio padrão e coeficientes de variância dos dois conjuntos de dados. As informações podem ser conferidas na Tabela 4.8.

O tempo decorrido entre a primeira tentativa até que ocorresse a primeira interação manteve-se constante e abaixo dos 10 minutos, como pode ser verificado na Figura 4.6. Isso mostra que, logo que o aluno submeteu sua resposta, o professor estava atento para o erro

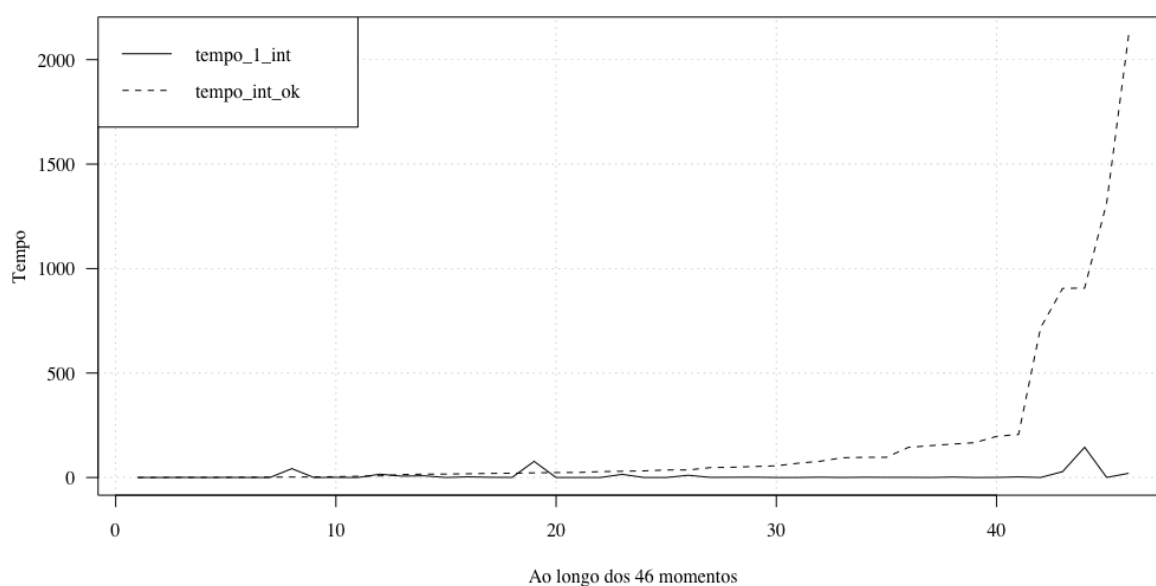


Tabela 4.8: Média, Desvio Padrão e Coeficiente de Variância do tempo

| <b>Tempo (em minutos)</b>          | <b>Média</b> | <b>DP</b> | <b>CV</b> |
|------------------------------------|--------------|-----------|-----------|
| Até primeira interação             | 8,45         | 24,67     | 291,8     |
| Da primeira interação até o acerto | 173,70       | 400,54    | 230,59    |

cometido e prontamente efetuou a intervenção necessária. Entretanto, após a intervenção do professor, os alunos demoraram mais tempo para acertar a resposta da questão. Em muitos casos, tal fato estava relacionado com a falta de domínio da sintaxe da linguagem de programação utilizada, como foi verificado na Seção 4.4.2.2.

Figura 4.6: Comparativo entre o tempo até a primeira interação e desde a interação até o acerto

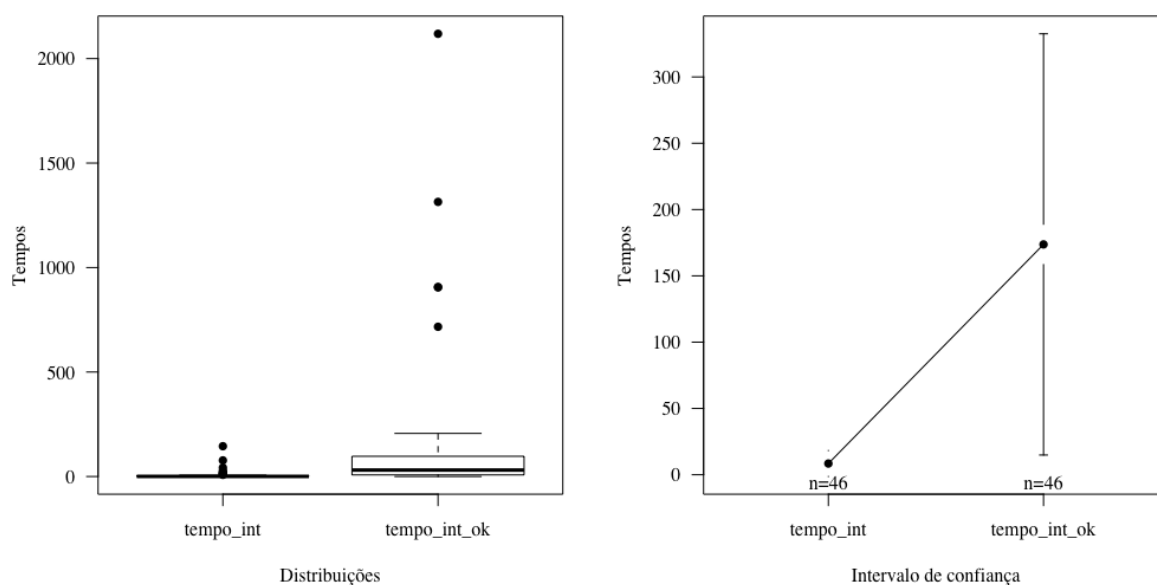


Houve casos, entretanto, em que o tempo gasto desde a primeira interação até o acerto foi extremamente elevado. Ao contrário da redução da quantidade de tentativas após a intervenção, confirmada na seção anterior, quando o tempo é observado, alguns casos apresentam intervalos bastante elevados. Em certos casos, conforme mostrado na Tabela 4.6, ainda que a primeira intervenção tenha ocorrido apenas há 17 minutos após a primeira tentativa, a conclusão acertada só ocorreu após 54 dias. É possível que, como não foi definido prazo para resolução dos exercícios, o aluno pode ter deixado as questões que teve

maior nível de dificuldade para depois.

Essa discrepância fica evidenciada quando a Figura 4.7 é observada. É possível verificar que há vários *outliers* na parte superior do lado direito do boxplot, referente ao tempo gasto após a intervenção até o acerto. Ao observar os intervalos de confiança, verifica-se que as duas médias encontram-se bem distantes uma da outra e que praticamente não há sobreposição entre os intervalos. Fica claro, porém, que houve uma demora bastante significativa desde a primeira interação até o acerto.

Figura 4.7: Boxplot e Plotmeans comparando os tempos até a primeira interação e desde a interação até o acerto



Após os testes de estatística descritiva, foram efetuados os testes necessários para verificar se os dados apresentam distribuição normal. Novamente foi utilizado o teste de Shapiro-Wilk, tomando como parâmetro que o *p-value* seja maior que o nível de significância  $\alpha$ , de 5%. O teste revelou, para os tempos desde a primeira tentativa até a primeira intervenção, um valor  $W = 0,38$  e  $p\text{-value} = 1,26 \times 10^{-12}$ . Com relação aos tempos decorridos desde a interação até o acerto foram obtidos os valores  $W = 0,46$  e  $p\text{-value} = 1,11 \times 10^{-11}$ .

Assim como ocorreu com os dados referentes às tentativas, os testes de normalidade

com os dados sobre os tempos nos retornaram valores de *p-value* ainda mais baixos. Tais resultados confirmam a não normalidade dos conjuntos de dados submetidos aos testes. É necessário, todavia, confirmar se a variância entre as médias dos dados apresenta ou não homogeneidade, por meio do Teste F. O teste apresentou como resultado  $F = 3,795 \times 10^{-3}$  e  $p\text{-value} < 2,2 \times 10^{-16}$ , valores extremamente baixos, que não confirmam a homogeneidade das variâncias entre os conjuntos de dados. Como os pressupostos de normalidade de homogeneidade não foram satisfeitos, é requerida a transformação dos dados, que foi efetuada pelo procedimento de Boxcox, com  $\lambda = 1,488$ .

Após a transformação dos dados, os testes para verificação de normalidade e homogeneidade foram retomados. Mesmo com a transformação dos dados, os resultados do Teste de Shapiro-Wilk continuaram com valores que não confirmavam que sua distribuição era normal. Para os dados sobre o tempo até a primeira interação, foi obtido  $W = 0,272$  e  $p\text{-value} = 9,251 \times 10^{-14}$ . Considerando os dados desde a interação até o acerto, os valores foram  $W = 0,355$  e  $p\text{-value} = 6,138 \times 10^{-13}$ .

Novamente foi efetuado novo teste para verificar a homogeneidade da variância após a transformação dos dados. O novo Teste F executado apresentou como resultados  $F = 2,919 \times 10^{-4}$  e  $p\text{-value} < 2,2 \times 10^{-16}$ . Como nem os pressupostos de normalidade nem de homogeneidade foram contemplados após a transformação dos dados, foi efetuado o método não paramétrico de Wilcoxon. Os resultados do teste não paramétrico mostraram  $W = 291$  e  $p\text{-value} = 2,148 \times 10^{-9}$ . Como pode ser verificado, os valores indicam que as médias apresentam diferenças entre elas. Conclui-se, portanto, que a quantidade de dados para os métodos estatísticos propostos são insuficientes, apontando para a necessidade da construção da metodologia, apresentada no Capítulo 5.

### 4.5.3 Quantidade de interações

Este capítulo teve como principal objetivo estudar a efetividade da interação promovida pela FARMA-ALG. A análise crítica e estatística das mensagens e comentários confirmam que a influência do uso da FARMA-ALG nos resultados é positiva. Todos os alunos que estavam com alguma dificuldade na resolução dos exercícios conseguiram avançar após

receber alguma intervenção do professor. Foi verificado, entretanto, que ocorreram poucas interações, considerando a quantidade de alunos e de respostas que foram analisadas.

Por mais eficaz que a plataforma seja em seu objetivo principal de auxiliar o aluno com dificuldade, o número de interações foi baixo. Em outras palavras, a FARMA-ALG é efetiva em seu propósito de remediar uma questão errônea por meio da intervenção docente, porém seu alcance tem sido baixo. As causas da reduzida quantidade de interações foram explicadas na Seção 4.4. Faz-se necessária, portanto, a elaboração de uma concisa e eficiente metodologia que permita a adoção da FARMA-ALG de modo que, para além de remediar o erro, o faça promovendo a interação entre professores e alunos, com a maior abrangência possível. Essa metodologia é proposta no próximo capítulo e tem como base, pressupostos e estratégias para promoção da interação em STIs.

## 4.6 Considerações finais sobre os estudos da interação

Este capítulo iniciou com o delineamento dos objetivos dos estudos da efetividade da interação que a FARMA-ALG promove. Posteriormente, foi estudada sua arquitetura e como ocorre o fluxo que permite a interação entre professores e alunos por meio da plataforma, com o auxílio de métodos e técnicas de estatística descritiva e inferencial. Nessa etapa, foi gerada uma documentação sobre os relacionamentos entre as tabelas do banco de dados da FARMA-ALG, bem como sobre o fluxo de respostas e comentários entre professores e alunos.

Foram efetuadas observações empíricas sobre a influência da plataforma nos resultados de aprendizagem dos alunos. Os resultados da influência do uso da ferramenta na assertividade das respostas mostraram que, após a intervenção do professor, o aluno que apresentava dificuldade conseguiu avançar na resolução do exercício. Conclui-se que é recomendável efetuar futuras análises com dados de mais turmas, para que seja possível confirmar com mais precisão a efetividade da interação promovida pela FARMA-ALG.

Foram efetuadas análises estatísticas utilizando métodos descritivos e inferenciais. Foi detectada uma redução de 46% na quantidade de tentativas após a primeira intervenção do professor. Em contrapartida, houve um aumento no tempo gasto pelo aluno para responder

corretamente após receber ajuda, o que sugere que, devidamente instruído e orientado, passou de um estado anterior de “tentativa e erro”, para respostas mais embasadas. Diante dos resultados dos estudos, constatou-se a necessidade de elaboração de documentação, com metodologia concisa e replicável, que possa auxiliar professores e alunos que utilizam a FARMA-ALG, para que todo o seu potencial sociointerativo seja efetivamente alcançado.

## CAPÍTULO 5

### PROPOSTA DE METODOLOGIA COM ENFOQUE NA INTERAÇÃO

A FARMA-ALG oferece vários recursos para que professores e alunos usufruam das vantagens de um mecanismo de aprendizado que prioriza a abordagem sociointeracionista. Mais que um ambiente em que o professor pode elaborar e disponibilizar os Objetos de Aprendizagem, ela proporciona a experiência da interação. Ainda que a interação ocorra de forma assíncrona, nela reside um dos componentes diferenciais da proposta da FARMA-ALG em, com base nas intervenções, contribuir para a melhoria do desempenho dos alunos.

Tecnicamente, nada impede, entretanto, que o professor somente crie e disponibilize os exercícios aos seus alunos, sem que haja nenhuma interação. A interação não é tratada como pré-requisito para que a plataforma possa ser utilizada, cabendo única e exclusivamente ao professor a decisão de interagir com seus alunos, com comentários nas questões que são submetidas. Todas as demais funcionalidades, como visualização, busca e manipulação, principalmente com o recurso visual do grafo de similaridade, são disponibilizadas de modo automático para que o professor possa fazer uso delas no sentido de ter apoio para identificar situações de similaridade entre respostas.

Conforme visto na Seção 4.4, as quatro turmas de 2016 receberam apenas duas intervenções, em contraste com as três turmas de 2015, que tiveram 80 comentários. Tal fato deve-se, em grande parte, à falta de um guia, uma metodologia que possa instruir os professores acerca dos procedimentos necessários para que seja promovida a interação. Considerando a importância da interação no processo de ensino-aprendizado que a FARMA-ALG pode proporcionar, foi detectada a necessidade em sistematizar uma metodologia para a utilização eficaz da plataforma. O foco é principalmente sobre os procedimentos necessários para promoção da interação entre professores e alunos.

Há poucas metodologias que auxiliam o ensino pelo modelo sociointeracionista, baseado na interação por meio da tecnologia. É importante atravessar essa limitação, para que professores e alunos possam utilizar novas ferramentas tecnológicas digitais sob um marco guia estruturado que permita explorar os benefícios e vantagens desta forma de interação (GARCIA, 2013). Nas próximas seções, são descritos os passos para a proposição de uma metodologia genérica baseada em um modelo de processos que permitirá guiar o professor na utilização da FARMA-ALG em um contexto em que a interação configura como importante elemento.

## **5.1 Pressupostos e estratégias para metodologias em ambientes interacionistas**

Para delinear o que deve constar em um guia para uma metodologia que atenda as necessidades de interação da FARMA-ALG, primeiramente é necessário identificar quais são os pressupostos que devem ser considerados para que o uso da plataforma contemple a proposta interacionista. Tomando o pressuposto de Vygotsky, a escola, representada aqui pelo professor, deve desempenhar a função de mediar e assim, promover a transformação necessária para o aprendizado (KUTZKE; DIRENE, 2015). A interação deve ser tratada como elemento fundamental para para que a adoção da FARMA-ALG obtenha pleno êxito.

### **5.1.1 Promoção do desenvolvimento - Conceito de ZDP**

A resolução de exercícios de algoritmos requer do aluno grande esforço no que concerne o uso do pensamento lógico e da capacidade de abstração de problemas reais. Alguns alunos conseguem resolver questões de programação com certa facilidade, mas a maioria necessita de auxílio inicial. Tal conjuntura é decorrente do histórico de dificuldades enfrentadas pelos alunos, com relação ao raciocínio lógico exigido por essas disciplinas, conforme já discutido na Seção 2.2.2. A interação na FARMA-ALG permite que o professor possa atuar como intermediador no processo de aquisição do conhecimento que eventualmente o aluno ainda não tenha, para que este possa prosseguir de modo independente nas próximas

questões.

Chickering e Ehrmann (1996) elencam sete princípios ou estratégias para uso efetivo de tecnologias da informação e comunicação para a educação:

1. encorajamento do contato entre alunos e professores;
2. desenvolvimento da reciprocidade e cooperação entre estudantes;
3. uso de técnicas de aprendizagem ativa;
4. fornecimento imediato de retorno ao aluno (*feedback*);
5. ênfase do tempo na tarefa;
6. criação de expectativas elevadas; e
7. respeito da diversidade de talentos e formas de aprendizagem.

Com base nas estratégias supracitadas, há estrutura suficiente para projetar uma metodologia estruturada em processos, que dê suporte às atividades mediadas pela FARMA-ALG.

As fases necessárias para que o aluno possa sair de um estágio de dependência da intervenção docente e consiga resolver problemas de forma independente são contempladas no conceito de Zona de Desenvolvimento Proximal (ZDP). A ZDP consiste “*na distância entre o nível de desenvolvimento atual determinado pela resolução independente de problemas e o nível de desenvolvimento potencial determinado pela resolução de problemas sob orientação ou em colaboração com parceiros mais capazes*”. (VYGOTSKY, 1987, p. 211). O conceito acima é um dos mais importantes na área da Pedagogia, sendo basilar para a presente metodologia. A capacidade do aluno em conseguir resolver problemas sem auxílio reduz na medida em que a dificuldade da tarefa aumenta, de acordo com a Figura 5.1.

Lui (2012) nos diz que, em um contexto social, a ZDP refere-se à área entre o nível de desenvolvimento atual e o nível desenvolvimento potencial. Também é conhecida como nível instrucional, que é onde a instrução deve ser priorizada para garantir os melhores ganhos de aprendizado para cada estudante. No nível abaixo, situa-se o nível de desenvolvimento atual, também referido como nível independente, que engloba habilidades que um aluno já dominou e pode realizar independentemente, entretanto as instruções aqui focadas não criariam um desafio suficiente para os alunos. O nível mais superior refere-se ao que o



aluno não é capaz de realizar de forma independente, mas que, com a ajuda do professor ou de colegas mais experientes, consegue resolver. O ensino não deveria ocorrer neste nível, pois não resultaria em um aprendizado efetivo.

Figura 5.1: Zona de Desenvolvimento Proximal



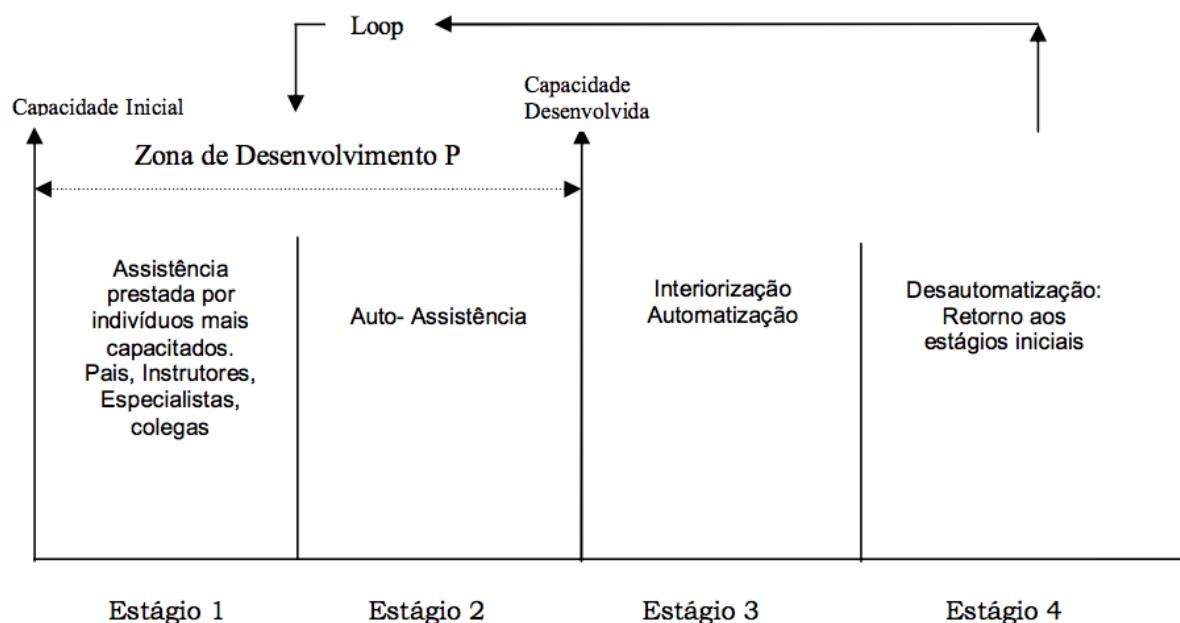
Fonte: (LUI, 2012). Traduzido e adaptado pelo autor.

### 5.1.2 Estágios da ZDP

Conforme pode ser verificado na Figura 5.2 a ZDP pode ser dividida em quatro estágios, cada um deles detalhadamente explicados a seguir, de acordo com Dunphy e Dunphy (2003). O Estágio 1 é onde o desempenho é acompanhado por outras pessoas que têm mais conhecimento. A quantidade de intervenção externa depende da natureza da tarefa e das características do aprendiz. Nos estágios iniciais, como o aluno pode ter um entendimento limitado, o professor oferece indicações ou modelos e as respostas do aluno tendem a ser imitativas. Somente de forma gradual o aluno compreenderá a maneira como as partes da atividade se relacionam entre si ou entenderá o significado do desempenho.

Quando algum conceito de desempenho global for adquirido, pode ser oferecida uma assistência adicional por meio de perguntas, *feedback* e estruturação cognitiva adicional. Os professores que auxiliam no desempenho dos alunos precisam ter um profundo conhecimento do assunto, para que ele possa classificar as tarefas em metas e sub-metas. Sem esse conhecimento, os professores não podem reformular rapidamente os objetivos da interação,

Figura 5.2: Estágios da Zona de Desenvolvimento Proximal



Fonte: (THARP; GALLIMORE, 1991)

pois eles não serão capazes de mapear a concepção que o aluno tem acerca do objetivo da tarefa sobre as estruturas superiores de conhecimento da disciplina que está sendo ensinada.

Durante o Estágio 1, há um declínio constante da responsabilidade do professor para o desempenho da tarefa e um aumento recíproco na proporção do aprendiz na responsabilidade. A tarefa de desenvolvimento da primeira fase é para permitir o trânsito de uma situação de regulação para auto-regulação. A tarefa do professor é adequar de forma precisa a assistência ao aluno, respondendo ao esforço do aluno e à compreensão da meta da tarefa. A tarefa da primeira etapa é realizada quando a responsabilidade pela adaptação da assistência, adaptando a transferência e executando a tarefa em si, foi efetivamente entregue ao aprendiz.

O Estágio 2 é onde o desempenho do aluno é promovido por ele mesmo, sendo capaz de realizar uma tarefa sem o auxílio de outros. No entanto, isso não significa que o desempenho esteja totalmente desenvolvido ou automatizado. Os processos de interiorização e automatização do Estágio 3 são trazidos para este estágio, passando o controle e/ou a assistência do professor ao aluno. O aluno que antes foi orientado por

outros está agora começando a ser orientado e dirigido por ele mesmo. Considerando o nível microgenético, os adultos constantemente conversam consigo mesmos e se ajudam de todas as formas possíveis.

O Estágio 3 é onde o desempenho é desenvolvido e automatizado. Uma vez que todas as evidências de auto-regulação desaparecerem, o aluno então emerge da ZDP para o estágio de desenvolvimento da tarefa. A execução da tarefa passa a ser suave e integrada, uma vez que ela foi internalizada e “automatizada”. A assistência do perito não é mais necessária, podendo ser, nesta fase, até mesmo perturbadora. Nessa condição as instruções dos outros são perturbadoras e irritantes pois nessa fase a autoconsciência pode ser prejudicial à integração suave de todos os componentes da tarefa. O desempenho já não está mais em desenvolvimento; ele está desenvolvido.

O Estágio 4 é onde a desautomatização do desempenho conduz à recursão para a ZDP. A aprendizagem de um indivíduo ao longo de sua vida é composta por essas mesmas sequências controladas dentro da ZDP, das assistências de terceiros para a auto-suficiência, repetindo recursivamente para o desenvolvimento de novas capacidades. Para cada indivíduo, em algum momento, haverá uma mistura da assistência de terceiros, de auto-assistência e de processos automatizados. Até mesmo aquele que já é considerado um perito competente pode aproveitar essa sequência para manter e melhorar seu desempenho. Na verdade, o aprimoramento, a melhoria e a manutenção do desempenho proporcionam um ciclo recorrente que vai da auto-assistência de volta para a assistência de terceiros.

### **5.1.3 A ZDP para promoção da interação**

Uma das possibilidades que a ZDP oferece refere-se à possibilidade de generalização de seu uso para qualquer tipo de conteúdo e para qualquer idade. Isto posto, é possível utilizar os pressupostos que permitem aplicar a ZDP para um conteúdo de aprendizagem que dependa da intervenção de outra pessoa mais competente e que exija que o aluno tenha um potencial de aprendizagem que lhe permita avançar no domínio da tarefa (GARCIA; DIRENE, 2013). Há três pressupostos que Chaiklin (2011) traz em seu trabalho, que elucidam o entendimento sobre a ZDP, que são os pressupostos da generalidade, da

assistência e do potencial, discutidos a seguir.

O pressuposto da generalidade assume que a ZDP não está ligada ao desenvolvimento de habilidades de alguma tarefa particular, mas do processo de desenvolvimento do indivíduo, indistintamente da particularidade da tarefa. Segundo o pressuposto da assistência, uma pessoa sem conhecimentos prévios sobre uma determinada especialidade sempre será capaz de fazer mais e resolver tarefas mais difíceis em colaboração, sob direção ou mediante auxílio do que de forma independente. De acordo com o pressuposto do potencial, este traz indícios da presença de certas funções que ainda estão em maturação, devendo não ser tratadas como uma propriedade já adquirida, mas como algo que será alvo de uma ação intervencionista significativa.

Como embasamento para a criação de uma metodologia de uso efetivo da FARMA-ALG pelos professores, foram considerados todos os pressupostos essenciais, com ênfase no pressuposto da assistência. Tais pressupostos auxiliam na formulação de processos bem definidos que serão a base da metodologia. Trabalhar na ZDP pode ajudar os professores a identificar pontos forte e fracos dos alunos, possibilitando que os conteúdos possam ser ministrados de acordo com o nível dos alunos, por meio do estímulo da interação social entre eles como forma de consolidar a aprendizagem. Os processos são elencados e detalhados na seção seguinte, sempre com o enfoque de ofertar ao professor um guia confiável para promoção da interação ao utilizar a FARMA-ALG.

## **5.2 Atividades do professor na FARMA-ALG**

Um ambiente informatizado que promova a mediação através do erro só terá seus objetivos plenamente alcançados se professores e alunos efetivamente utilizarem os recursos de interação oferecidos pela ferramenta. Por melhor e eficiente que seja o mecanismo de envio de comentários, caso o professor não o utilize, o intuito sociointeracionista da ferramenta torna-se nulo. Caso isso ocorra, seu uso fica restrito somente aos aspectos funcionais de criação e disponibilização de Objetos de Aprendizagem, a submissão de repostas pelos alunos e sua correção semi-automatizada. Mesmo que a o professor tenha a sua disposição os recursos de visualização e manipulação das repostas por meio do Grafo

de Similaridade, a real interação com seus alunos somente acontecerá se houver o envio de mensagens e comentários.

Segundo Hannafin (1992 apud GARCIA; DIRENE, 2013), falta uma estrutura guia sobre como o professor deve utilizar as ferramentas tecnológicas digitais para alcançar melhorias no processo de ensino-aprendizagem. Em se tratando especificamente do uso da FARMA-ALG, não há nenhum método ou guia que auxilie o professor no sentido de orientá-lo para que faça uso da mediação que a ferramenta proporciona. É necessário, portanto, propor e descrever uma sequência de processos estruturados direcionados para a prática docente com o objetivo de guiar o professor para que a mediação efetivamente ocorra.

A utilização de uma ferramenta como a FARMA-ALG exige do professor organização e planejamento para que as aulas sejam eficientes e produzam o resultado desejado. São elencadas a seguir quais funções e atividades o professor deve executar para uso efetivo da ferramenta:

1. identificar qual conteúdo da disciplina será ministrado;
2. ministrar as aulas, de acordo com o método de sua preferência;
3. elaborar os Objetos de Aprendizagem relativos ao conteúdo, contendo:
  - nome e
  - descrição;
4. editar os OAs criando:
  - introdução,
  - exercícios,
  - questões; e
  - casos de teste;
5. efetuar as correções dos exercícios propostos na FARMA-ALG;
6. identificar alunos com níveis de dificuldade que requeiram intervenção;
7. estabelecer a interação com esses alunos, por meio de comentários e mensagens;
8. acompanhar a evolução dos alunos após a intervenção.

### 5.3 Fases da metodologia

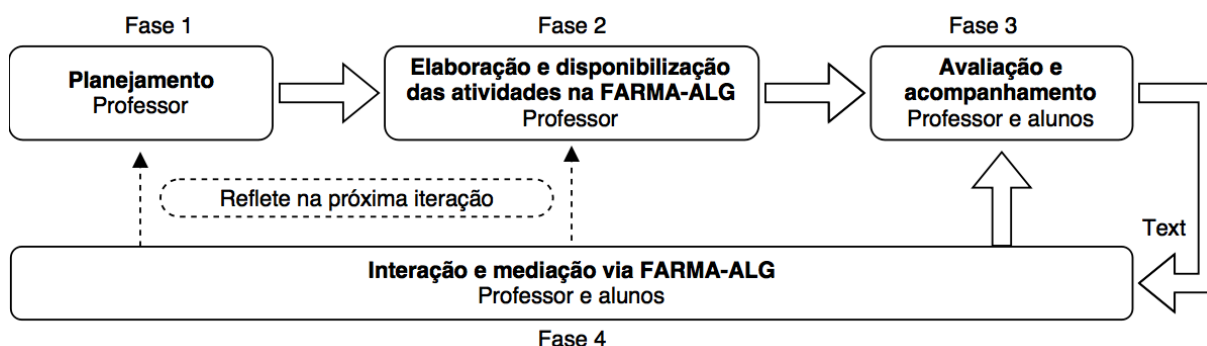
As atividades elencadas anteriormente são imprescindíveis para o uso efetivo da FARMA-ALG, entretanto, elas ainda não estão organizadas em um conjunto estruturado que configure uma metodologia que possa ser replicada. Uma atenção especial deve ser dedicada às atividades de correção, identificação de alunos com dificuldades, interação e acompanhamento dos mesmos. Nesta seção, são descritos os processos que a metodologia deve conter para contemplar as necessidades de orientar o professor na implementação bem sucedida do modelo interacionista proposto pela plataforma FARMA-ALG. Considera-se que os alunos e professores já tenham o conhecimentos necessário para operar a plataforma.

As atividades são distribuídas entre quatro fases lógicas, conforme segue:

1. planejamento (professor),
2. elaboração e disponibilização das atividades (professor),
3. avaliação e acompanhamento (professor e alunos); e
4. interação e mediação (professor e alunos).

Nesse modelo, as atividades de cada fase precisam ser completadas para que se avance para a próxima, de acordo com a descrição da Figura 5.3. A estrutura proporciona um mecanismo iterativo, de modo que o professor pode voltar às fases anteriores e utilizar as informações para melhorar as atividades e atingir os objetivos propostos.

Figura 5.3: Fases da metodologia



Os objetivos de cada uma das fases e os produtos que devem ser entregues estão relacionados na Tabela 5.1 e são detalhados em passos nas seções a seguir.

Tabela 5.1: Objetivos e produtos das fases

| <b>Fase</b> | <b>Objetivo da fase</b>  | <b>Produto da fase</b>                                    | <b>Atores</b>      |
|-------------|--|---|--------------------|
| Fase 1      | Definição dos objetivos da disciplina                                  | Plano de ensino   | Professor          |
| Fase 2      | Elaboração e disponibilização dos Objetos de Aprendizagem na FARMA-ALG | Objetos de Aprendizagem contendo exercícios e questões    | Professor          |
| Fase 3      | Avaliação e acompanhamento das dificuldades dos alunos                 | Relatório apontando as principais dificuldades dos alunos | Professor e alunos |
| Fase 4      | Interação e mediação com os alunos                                     | Intervenção com os alunos e adequação dos exercícios      | Professor e alunos |

### 5.3.1 Planejamento

A fase de planejamento tem como principal meta definir quais são os objetivos de aprendizagem que a disciplina deve oferecer. Tais objetivos descrevem quais conceitos e habilidades que os alunos devem ter conhecimento e compreensão para que estejam habilitados a analisar problemas, propor soluções e implementá-las. Essa fase é subdividida em três etapas, nas quais o professor identificará as necessidades e requisitos, definirá quais serão os objetivos de aprendizagem e analisará como deve ser o uso da estrutura informatizada disponível.

#### 5.3.1.1 Etapa 1: Identificação das necessidades e requisitos

1. As competências que os alunos devem adquirir ao concluir as atividades propostas devem ser identificadas. Uma boa base para as escolhas é a ementa da disciplina, normalmente disponibilizada pela coordenação do curso, bem como seu projeto pedagógico.
2. As linguagens de programação que serão utilizadas devem ser as mesmas às quais a FARMA-ALG oferece suporte. Com base nessa informação, o professor deverá indicar aos alunos um guia de referência da linguagem adotada, com o intuito de dirimir eventuais erros básicos de sintaxe.
3. Por se tratar de disciplina que exige bastante das habilidades de abstração e

cognição de aspectos sobre lógica, é interessante efetuar uma avaliação no sentido de diagnosticar eventuais discrepâncias de conhecimentos. Ao realizar esse procedimento, esta avaliação prévia permite saber qual é o Nível de Desenvolvimento Atual, abordado na Seção 5.1.1.

### **5.3.1.2 Etapa 2: Definição dos objetivos de aprendizagem**

A partir dos resultados observados na avaliação de diagnóstico, na etapa anterior, o Nível de Desenvolvimento Atual estará devidamente identificado. Com base nessa informação, os seguintes passos devem ser realizados:

1. tomando a lista de competências obtidas na Etapa 1, identifica-se quais as perícias que o aluno deve adquirir ao final da avaliação;
2. de acordo com as perícias identificadas no passo anterior, define-se os conteúdos que devem estar contidos nos OAs; e
3. redigir, com base nos passos anteriores, uma lista formal dos objetivos de aprendizagem.

### **5.3.1.3 Etapa 3: Análise da viabilidade em alcançar os objetivos de aprendizagem utilizando a FARMA-ALG**

1. Avaliar todos os objetivos de aprendizagem e identificar, de acordo com o nível de complexidade de cada um, quais podem ser transformados em Objetos de Aprendizagem para que possam ser disponibilizados na FARMA-ALG.
2. Os objetivos que não se encaixam na possibilidade acima devem ser tratados com abordagem alternativa à plataforma.

## **5.3.2 Elaboração e disponibilização das atividades**

Nessa fase o professor irá, com base na ementa da disciplina, elaborar Objetos de Aprendizagem que serão disponibilizados como atividades na FARMA-ALG. Essas atividades são compostas por questões de algoritmos, organizadas em listas de exercícios cujo nível de dificuldade aumenta gradualmente. Nas três etapas que compõem essa fase,



o professor irá elaborar as atividades, verificar a aderência das atividades à proposta e recursos ofertados pela FARMA-ALG e definir o plano de trabalho com o cronograma de sessões.

### **5.3.2.1 Etapa 1: Escolha dos exercícios**

1. De acordo com o que foi obtido na definição dos objetivos de aprendizagem, na Etapa 2 da fase anterior, selecionar o conteúdo que será trabalhado.
2. Estabelecer quantos exercícios serão necessários para alcançar o objetivo.
3. Definir quais exercícios devem compor a primeira lista para sabermos qual é o nível atual de desenvolvimento.
4. Definir em quantas listas os demais exercícios serão divididos e como será a distribuição entre as listas, conforme a dificuldade dos exercícios. Cada lista deve conter exercícios cuja dificuldade permita que os alunos atinjam seu nível de desenvolvimento potencial.

### **5.3.2.2 Etapa 2: Elaboração dos Objetos de Aprendizagem**

1. Transformar cada exercício em um Objeto de Aprendizagem compatível com os pré-requisitos da FARMA-ALG.
2. Distribuir os exercícios em listas, de acordo com a quantidade de níveis que o professor definiu ser necessária para que o aluno conclua com sucesso uma lista e esteja preparado para a próxima.
3. Elaboração dos casos de teste, considerando a necessidade de detecção de possíveis *bugs* nas respostas submetidas pelos alunos.
4. Disponibilização dos casos de teste na plataforma.

### **5.3.2.3 Etapa 3: Elaboração do cronograma das atividades**

1. Elaborar cronograma de disponibilização online de cada lista de exercícios, conforme andamento da disciplina em sala de aula.
2. Definir a realização das atividades de acordo com a ordem das listas de exercícios,

sua duração e a data em que deverão ser respondidas. Para tanto, é necessário revisar disponibilidade de tempo do professor, monitores da disciplina e dos alunos.

3. Estipular prazos para resolução dos exercícios propostos.
4. Indicar quem serão os monitores das disciplinas e informar a turma.

### **5.3.3 Avaliação e acompanhamento**

Na fase de avaliação e acompanhamento, o professor, em conjunto com o monitor da disciplina, efetuará a correção das respostas dos exercícios submetidas pelos alunos na FARMA-ALG. Durante a correção, é fundamental que sejam identificadas as principais dificuldades enfrentadas pelos alunos. Ao final dessa fase, deverá ser gerado um relatório que indique quais foram os aspectos que se apresentaram como dificuldades na resolução dos exercícios.

#### **5.3.3.1 Etapa 1: Preparação da sessão**

1. Compor um manual de instruções de uso da FARMA-ALG e disponibilizá-lo na própria plataforma, preferencialmente na tela inicial. Deverá constar quais são os requisitos técnicos de hardware, de software e de rede para que os usuários tenham acesso.
2. Publicar os períodos em que cada lista de exercícios estará disponível.
3. Informar aos alunos e ao monitor, via e-mail, o endereço para acessar a plataforma, bem como os procedimentos necessários para efetuar o cadastro e matrícula na disciplina.
4. Definir como os alunos obterão suporte em caso de problemas com a plataforma.
5. Disponibilizar os OAs para que os alunos matriculados possam acessar as listas de exercícios e resolver as questões propostas.

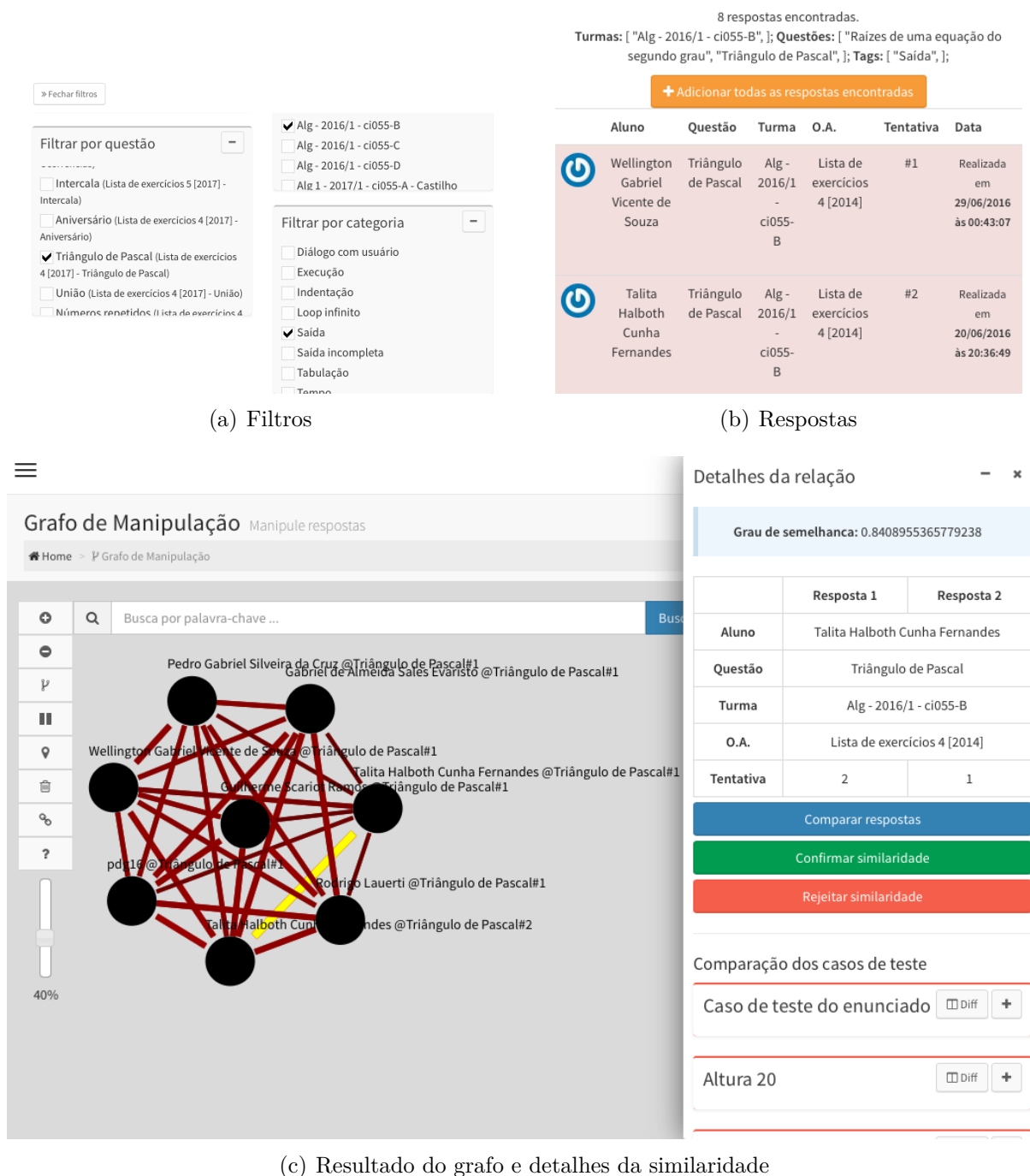
### 5.3.3.2 Etapa 2: Desenvolvimento da sessão

1. Antes de efetivamente liberar a plataforma para uso, efetuar testes de conexão para diagnosticar eventuais problemas em tempo hábil para solucioná-los.
2. Liberar o acesso à FARMA-ALG aos alunos, somente para a lista de exercícios cujo conteúdo foi previamente desenvolvido em sala de aula.
3. Periodicamente, acompanhar a participação dos alunos, verificando se estão conseguindo acessar as questões e submetendo suas repostas sem nenhum tipo de problema.

### 5.3.3.3 Etapa 3: Acompanhamento das respostas durante o prazo de submissão

1. No painel principal da FARMA-ALG, verificar os seguintes itens:
  - anotar a quantidade de respostas incorretas;
  - verificar se há comentários e mensagens enviadas pelos alunos, direcionados ao professor ou monitor; e
  - verificar a efetividade dos casos de teste e efetuar correções quando necessário.
2. No menu “Grafo de Manipulação”, com o uso adequado dos filtros disponíveis, efetuar as seguintes ações:
  - configurar os filtros para que a busca seja efetuada para uma questão, de uma turma e que esteja em uma determinada categoria de erro. Ex.: Questão: Triângulo de Pascal; Turma: Alg - 2016/1; Categoria: Saída. Vide exemplo na Figura 5.4; e
  - Identificar as respostas incorretas com o maior grau de similaridade entre elas.
3. Usando o menu “Linha do Tempo”, efetuar procedimento similar ao do passo anterior, com o objetivo de identificar respostas com o maior grau de similaridade possível.
4. Adicionalmente, utilizar os recursos de “Busca” para encontrar padrões de erros.
5. Utilizar, caso necessário o recurso de “Tags”, que permite selecionar as repostas com determinados tipos de erros, conforme a Figura 5.5.

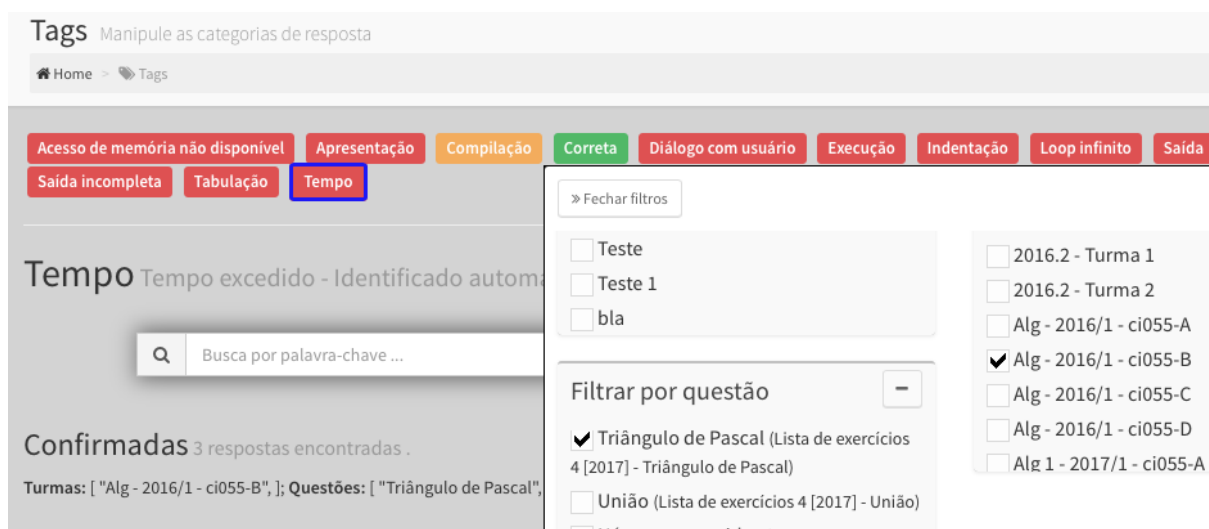
Figura 5.4: Filtros e resultado de similaridade de questões



### 5.3.4 Interação e mediação

Nesta fase, ocorre a interação entre alunos, monitores e professores. Conforme as questões vão sendo respondidas, professores e monitores devem identificar os alunos que enfrentaram dificuldades. A observação da similaridade em respostas incorretas, efetuada na fase anterior, deve ser utilizada para subsidiar essa tarefa. Em seguida, o professor e o

Figura 5.5: Resultado da busca por tags de erros



monitor devem promover a interação com os alunos, por meio do envio de comentários específicos sobre os erros mais comuns, cometidos nas tentativas de resolução dos exercícios, baseados nos tipos de erros que foram vistos na Seção 4.4.2.2. Caso necessário, além de conversar com o aluno, para que seja possível compreender sua dificuldade e auxiliá-lo, pode ser necessária a adequação do exercício.

### 5.3.4.1 Etapa 1: Identificação dos alunos que necessitam de apoio

1. Ler as mensagens e comentários enviados pelos alunos. Se eles as enviaram antes do professor, significa que estão solicitando esclarecimentos.
2. Identificar os alunos que cometeram erros, de acordo com os seguintes critérios:
  - nível de progresso na resolução das questões da lista de exercícios vigente. Quanto menor for a barra de progresso, maior a possibilidade de que o aluno esteja enfrentando dificuldades e necessitando de ajuda;
  - número de tentativas por Objeto de Aprendizagem. Listas de exercícios com grande quantidade de tentativas podem indicar que os alunos não estão seguros de que suas respostas estão corretas; e
  - alunos que cometeram erros muito similares com os de outros colegas. Os dados

obtidos na Etapa 3 da Fase anterior são utilizados para este fim.

#### **5.3.4.2 Etapa 2: Identificação das questões que os alunos tiveram mais dificuldade em solucionar**

1. Identificar as questões que obtiveram maiores quantidade de respostas incorretas.
2. Categorizar as questões incorretas pelos tipos de erros, apresentados na Seção 4.4.2.2.

#### **5.3.4.3 Etapa 3: Intervenção com os alunos**

1. Responder mensagens e comentários dos alunos que já as enviaram.
2. Enviar mensagens ou comentários aos grupos de alunos com dificuldades similares em determinadas questões.
3. Enviar mensagens ou comentários individuais aos alunos que estiverem com alguma dificuldade específica.
4. Acompanhar o desempenho dos alunos após a intervenção.
5. Complementar a intervenção com mensagens e comentários adicionais, caso seja necessário.

### **5.4 Considerações finais sobre a metodologia**

Quanto mais simples e fácil for a utilização de uma plataforma, mais fácil será sua adoção. Entretanto, mesmo que os aspectos de usabilidade sejam visíveis e favoráveis, há outros que, apesar de importantíssimos, não estão explicitamente declarados e documentados na FARMA-ALG. Como exemplo, há o recurso de mediação do erro por meio da interação, um dos principais diferenciais da ferramenta.

A falta de uma metodologia, caso não haja auxílio de um professor com experiência na utilização da FARMA-ALG, pode comprometer a amplitude e efetividade das ações interacionistas propostas. Com a formalização e disponibilização de uma metodologia que funcione como um guia em todas as fases do processo de ensino, professores e alunos podem se beneficiar dos recursos de interação com mais facilidade. Espera-se, desse modo,

que todas as características sociointeracionistas que a FARMA-ALG oferece, possam ser aproveitadas em sua plenitude.

Ainda que professor tenha a orientação necessária, fornecida pela metodologia, é importante que o aluno utilize a FARMA-ALG de modo correto. É fundamental fornecer, sob a perspectiva do aluno, orientações que abordem de forma abrangente e concisa, as funcionalidades da ferramenta, para que sua utilização seja efetiva. Nesse sentido, registra-se a necessidade imprescindível em elaborar uma documentação da FARMA-ALG direcionada especificamente aos alunos.

## CAPÍTULO 6

### CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS

Neste trabalho, foi apresentada um estudo da FARMA-ALG, uma ferramenta de mediação do erro em disciplinas de algoritmos. Foi avaliado o quanto essa ferramenta tem contribuído para a melhoria do processo de ensino-aprendizagem. Para tanto, em primeiro lugar, fez-se um estudo sobre o uso da informática como ferramenta de apoio à educação. Foram buscados embasamento teórico sobre ambientes de autoria, objetos de aprendizagem, a questão do erro escolar e o histórico de sistemas e mecanismos de suporte ao erro, que considerassem a interação e o desenvolvimento cognitivo.

Foram abordados os objetivos do ensino de programação de computadores e, neste processo, a importância da disciplina de algoritmos. Ao discutir a disciplina de algoritmos, foi evidenciada a problematização da desistência e evasão, suas origens e consequências para a futura carreira do estudante. A partir desse problema, foi contextualizado o surgimento e a evolução de sistemas de diagnósticos de erros e posteriormente os sistemas de mediação de erros escolares. Para avaliar a ferramenta, foram efetuadas análises dos dados gerados, no formato de questões e seus comentários, utilizando as abordagens de (1) avaliação, (2) suporte e apoio ao estudante e (3) currículo, domínio de conhecimento e suporte aos professores.

Na sequência, foi apresentada em maiores detalhes, a FARMA-ALG, seu histórico e arquitetura, além de suas principais funcionalidades. Sob a perspectiva de um usuário docente, foi realizada uma análise das funcionalidades da plataforma, para verificação dos aspectos de interface e de experiência do usuário. Também foram propostas possíveis melhorias para a ferramenta, com a possibilidade de adição de novas funcionalidades, abordadas ao final da Seção 6.2.

A efetividade da interação promovida pela FARMA-ALG foi avaliada de três formas distintas. Primeiramente, foi avaliada a influência da ferramenta nos resultados das



respostas submetidas pelos alunos às questões propostas. Em seguida, foram analisados, de modo empírico e crítico, os conteúdos e teores das mensagens trocadas entre professores e alunos para avaliar de que modo ocorreu a interação e como ela contribuiu para o desempenho dos alunos. Posteriormente, foram efetuadas as análises estatísticas, descritivas e inferenciais, sobre os conjuntos de dados das respostas obtidas.

Ao verificar o reduzido índice de interação, provocado pela baixa quantidade de mensagens enviadas pelos professores, foi proposta uma metodologia de uso da plataforma. A proposta contempla pressupostos e estratégias utilizadas em ambientes interacionistas, visando a promoção do desenvolvimento, com base no conceito de Zona de Desenvolvimento Proximal. A ZDP serviu portanto, como alicerce para a elaboração de uma metodologia que tomasse a promoção da interação como base para sua construção. Para tanto, foram identificadas as atividades que o professor deve desempenhar ao adotar e utilizar a FARMA-ALG como ferramenta de auxílio para o ensino de algoritmos.

## 6.1 Resultados alcançados

Nas análises estatísticas, foram utilizadas técnicas e métodos descritivos e inferenciais. Sob o ponto de vista da análise estatística descritiva, observou-se que houve uma redução de 46% na quantidade de tentativas após o alunos receber algum tipo de intervenção do professor, embora o tempo desde a primeira interação até o acerto tenha aumentado. Sobre a efetividade promovida pela FARMA-ALG, foi verificado que seu uso promoveu um aumento na quantidade de respostas corretas em cerca de 7,5%, considerando os desempenhos dos alunos, de semestre para outro. A comparação entre semestres foi possível porque os conteúdos foram aplicados pelos mesmos professores e seguindo a mesma ementa. Entretanto, foi verificado que ainda assim, o percentual de respostas incorretas continuou elevado, próximo de 70% da totalidade das respostas. A maioria das respostas incorretas ocorreram em questões das primeiras listas de exercícios, o que denota possível falta de domínio dos princípios básicos das linguagens utilizadas. Tal afirmação tem como base o fato de que a maior parte dos erros detectados foram de saída não esperada e de compilação.

Foi observado que o nível de interação, verificado pela quantidade de mensagens e comentários enviados pelos professores, foi extremamente baixo, alcançando apenas 1,93% do total de respostas. Apesar da reduzida quantidade de interações, foi verificado que elas foram essenciais para que os alunos conseguissem progredir na resolução dos exercícios. Entretanto, foi constatado que a elevada quantidade de tentativas dos alunos nas questões mais básicas pode ser reduzida mediante a antecipação da intervenção.

As questões que receberam comentários e mensagens enviados pelos professores foram cuidadosamente verificadas. Foi constatada grande quantidade de tentativas consecutivas de um mesmo aluno para uma única questão, o que denota possível situação de “tentativa e erro”. Diante desse quadro, foi observado que os comentários dos professores sempre tinham como objetivo orientar os alunos sobre onde estavam cometendo os erros e quais as formas de contorná-los. A intervenção do professor obteve sucesso ao verificar que os alunos que receberam alguma recomendação obtiveram sucesso em cerca de 65% do total das questões nas quais haviam cometido erros.

De acordo com o observado, a intervenção do professor na FARMA-ALG constitui grande contribuição para que os alunos possam sanar suas dúvidas e progredir na resolução dos problemas propostos. Considerando os testes de estatística inferencial, a quantidade reduzida de dados disponíveis não foi suficiente que pudessem ser feitas afirmações sobre a efetividade das ações, pois se constatarem distribuições não normais em todos os conjuntos de dados analisados.

Foi observada reduzida quantidade de interações, causadas em grande parte pela falta de um método que guie o professor em suas atividades com a FARMA-ALG. Nesse sentido, foi proposta uma metodologia de utilização da ferramenta com forte foco na interação. A proposta tem como base os pressupostos definidos pelo conceito de ZDP e foi concebida para auxiliar o professor em suas atividades de modo que promova o aumento da interação.

## 6.2 Trabalhos futuros

Nesta seção são elencadas algumas metas para trabalhos futuros e proposições de novas funcionalidades, baseadas na continuidade da presente pesquisa. Também é proposto o

aprofundamento nas questões sobre correção automática de respostas e geração de casos de teste, que não foram pesquisados neste trabalho por questões de tempo e escopo. Estas sugestões podem ser vistas a seguir:

- a quantidade de dados disponíveis para esta pesquisa foi consideravelmente reduzido, restringindo-se a 229 alunos distribuídos entre 7 turmas. Pesquisas envolvendo mais turmas, incluindo de outras instituições de ensino, podem revelar informações relevantes;
- pesquisas envolvendo maior número de alunos podem permitir que sejam efetuados testes com turmas controle e turmas experimentais, possibilitando que comparações mais precisas sejam realizadas;
- a quantidade de professores, caso seja maior, pode mostrar as diferenças entre suas formas de interação e seus resultados;
- considerando o contexto de funcionalidade da FARMA-ALG, salienta-se que atualmente todo o processo de construção de casos de teste deve ser efetuado manualmente pelo professor, no momento em que ele cria os objetos de aprendizagem. A automatização do processo de geração de bons casos de testes a partir de um código-fonte “oráculo”, pode poupar o professor da tarefa manual e demorada da geração de casos de teste, permitindo que ele invista mais tempo no seu trabalho. Há vasto campo para futuros trabalhos no sentido de automatizar a criação de bons casos de testes possam contribuir para a melhoria da FARMA-ALG;
- a possibilidade do próprio sistema verificar e apontar os erros cometidos em trechos de código pode ajudar o professor, poupando seu tempo. Uma arquitetura acoplável ao FARMA-ALG que faça verificações automáticas nos códigos submetidos pelos alunos e/ou que gere casos de teste automaticamente certamente constitui um amplo campo de pesquisas.

## BIBLIOGRAFIA

- ADAM, A.; LAURENT, J.-P. Laura, a system to debug student programs. Artificial Intelligence, Elsevier, v. 15, n. 1, p. 75–122, 1980. Citado na página 27.
- AINSWORTH, S. Deft: A conceptual framework for considering learning with multiple representations. Learning and Instruction, Elsevier, v. 16, n. 3, p. 183–198, 2006. Citado 3 vezes nas páginas 24, 27 e 30.
- AINSWORTH, S.; FLEMING, P. Evaluating authoring tools for teachers as instructional designers. Computers in human behavior, Elsevier, v. 22, n. 1, p. 131–148, 2006. Citado na página 22.
- ALEVEN, V.; SEWALL, J. Hands-on introduction to creating intelligent tutoring systems without programming using the cognitive tutor authoring tools (CTAT). In: INTERNATIONAL SOCIETY OF THE LEARNING SCIENCES. Proceedings of the 9th International Conference of the Learning Sciences-Volume 2. [S.l.], 2010. p. 511–512. Citado na página 23.
- ANDERSON, J. R. The architecture of cognition. [S.l.]: Psychology Press, 2013. Citado na página 16.
- BOULAY, B. du; SOTHCOTT, C. Intelligent systems for teaching programming. Artificial Intelligence Tools in Education. P. Ercoli and R. Lewis, Elsevier Science (North-Holland), 1988. Citado na página 27.
- BROWN, A. Metacognition, executive control, self-regulation, and other more mysterious mechanisms. Metacognition, motivation, and understanding, Erlbaum, p. 65–116, 1987. Citado na página 22.
- CABADA, R. Z.; ESTRADA, M. L. B.; GARCÍA, C. A. R. Educa: A web 2.0 authoring tool for developing adaptive and intelligent tutoring systems using a kohonen network. Expert Systems with Applications, Elsevier, v. 38, n. 8, p. 9522–9529, 2011. Citado na página 23.
- CASSANDRE, M. P.; QUEROL, M. A. P. Metodologias intervencionistas: Contribuição teórico-metodológica vigotskianas para aprendizagem organizacional/interventionist methodologies: Vigotskian theoretical and methodological contributions in organizational learning. Revista Pensamento Contemporâneo em Administração, Universidade Federal Fluminense, Faculdade de Administração, Ciências Contábeis e Turismo, v. 8, n. 1, p. 17, 2014. Citado na página 16.
- CHAIKLIN, S. A zona de desenvolvimento próximo na análise de Vigotski sobre aprendizagem e ensino. Psicologia em Estudo, SciELO Brasil, v. 16, n. 4, p. 659–675, 2011. Citado na página 104.
- CHICKERING, A. W.; EHRMANN, S. C. Implementing the seven principles: Technology as lever. AAHE bulletin, American Association for Higher Education, v. 49, p. 3–6, 1996. Citado 2 vezes nas páginas 26 e 101.

DIRENE, A. et al. Objetos de aprendizagem generalizáveis para o currículo de matemática do ensino médio. In: Anais do Workshop de Informática na Escola. [S.l.: s.n.], 2009. v. 1, n. 1, p. 1693–1702. Citado na página 24.

DIRENE, A. I. Designing intelligent systems for teaching visual concepts. International Journal of Artificial Intelligence in Education, v. 8, n. 1, p. 44–70, 1997. Citado na página 22.

DUARTE, N. Vigotski e o “aprender a aprender”: crítica às apropriações neoliberais e pós-modernas da teoria vigotskiana. [S.l.]: Autores Associados, 2001. Citado na página 22.

DUNPHY, B. C.; DUNPHY, S. L. Assisted performance and the zone of proximal development (ZPD): a potential framework for providing surgical education. Australian Journal of Educational & Developmental Psychology, v. 3, n. 2003, p. 48–58, 2003. Citado na página 102.

ESTANISLAU, E. A. Mediação pedagógica na EAD: o papel do professor nos fóruns de discussão do AVA MOODLE. 2014. Citado 2 vezes nas páginas 29 e 30.

FERREIRA, B. d. J. P.; DUARTE, N. O lema aprender a aprender na literatura de informática educativa. Educação e Sociedade, p. 1019–1035, 2012. Citado na página 21.

FEUERSTEIN, R. The theory of mediated learning experience: About human as a modifiable being. Ministry of Defense Publications, Jerusalem, 1998. Citado na página 31.

FIGUEIRA, A. P. C. Metacognição e seus contornos. Revista Iberoamericana de educación, n. 1681-5653, 2003. Citado na página 22.

FLÔRES, M. L. P.; TAROUÇO, L. M. R. Diferentes tipos de objetos para dar suporte a aprendizagem. RENOTE, v. 6, n. 2, 2008. Citado na página 15.

GARCIA, L. F. U. Metodologia para implementação de estratégias colaborativas mediadas por ferramentas de interação síncronas. Tese (Doutorado) — Universidade Federal do Paraná, 2013. Citado na página 100.

GARCIA, L. F. U.; DIRENE, A. Metodologia para implementação de estratégias colaborativas mediadas por ferramentas de interação síncrona. Tecnologias, Sociedade e Conhecimento, v. 1, n. 1, p. 80–99, 2013. Citado 3 vezes nas páginas 34, 104 e 106.

GIRAFFA, L. M. M.; MORA, M. d. C. Evasão na disciplina de algoritmo e programação: Um estudo a partir dos fatores intervenientes na perspectiva do aluno. In: III Conferencia Latinoamericana sobre abandono en la educación superior. Mexico. [S.l.: s.n.], 2013. Citado na página 28.

GIRAFFA, L. M. M.; VICARI, R. M. Estratégias de ensino em sistemas tutores inteligentes modelados através da tecnologia de agentes. Revista Brasileira de Informática na Educação. Florianópolis, 1999. Citado na página 34.

GOMES, A.; HENRIQUES, J.; MENDES, A. Uma proposta para ajudar alunos com dificuldades na aprendizagem inicial de programação de computadores. Educação, Formação & Tecnologias-ISSN 1646-933X, v. 1, n. 1, p. 93–103, 2008. Citado na página 28.

HANNAFIN, M. J. Emerging technologies, ISD, and learning environments: Critical perspectives. Educational Technology Research and Development, Springer, v. 40, n. 1, p. 49–63, 1992. Citado na página 106.

HASEMER, T. An empirically-based debugging system for novice programmers. Tese (Doutorado) — Open University, 1983. Citado na página 27.

ISOTANI, S. et al. Can erroneous examples help middle-school students learn decimals? In: Towards Ubiquitous Learning. [S.l.]: Springer, 2011. p. 181–195. Citado na página 27.

JENSEN, K.; WIRTH, N. PASCAL User Manual and Report. New York, NY, USA: Springer-Verlag New York, Inc., 1974. ISBN 3-540-06950-X. Citado na página 48.

JOHNSON, W. L.; SOLOWAY, E. Proust: Knowledge-based program understanding. Software Engineering, IEEE Transactions on, IEEE, n. 3, p. 267–275, 1985. Citado na página 27.

KERNIGHAN, B. W. The C Programming Language. 2nd. ed. [S.l.]: Prentice Hall Professional Technical Reference, 1988. ISBN 0131103709. Citado na página 48.

KOOHANG, A.; HARMAN, K. Learning objects: Theory, praxis, issues, and trends. [S.l.]: Informing Science, 2007. v. 2. Citado na página 15.

KRYNSKI, E. M. Uma abordagem metacognitiva através de múltiplas representações externas para o ensino de programação de computadores. Tese (Doutorado) — Universidade Federal do Paraná, Setor de Ciências Exatas, Programa de Pós-Graduação em Informática, 2007. Citado na página 25.

KUTZKE, A. R.; DIRENE, A. Mediação do erro na educação: um arcabouço de sistema para a instrumentalização de professores e alunos. In: Anais do Simpósio Brasileiro de Informática na Educação. [S.l.: s.n.], 2014. v. 25, n. 1, p. 737–746. Citado 5 vezes nas páginas 17, 27, 32, 36 e 41.

KUTZKE, A. R.; DIRENE, A. I. Informática educacional e a mediação do erro na educação : um estudo teórico-crítico e uma proposta de instrumento computacional. [S.l.]: 2015., 2015. Citado 7 vezes nas páginas 17, 18, 25, 37, 38, 39 e 100.

LEITE, M. et al. Otimizando o processo de ensino e aprendizagem com a arquitetura para desenvolvimento de objetos de aprendizagem-adoa. In: Anais do Simpósio Brasileiro de Informática na Educação. [S.l.: s.n.], 2015. v. 26, n. 1, p. 1002. Citado na página 15.

LEITE, M. D.; PIMENTEL, A. R. Arquitetura para remediação de erros baseada em múltiplas representações externa. [S.l.]: 2013., 2013. Citado na página 30.

LEITE, M. D.; PIMENTEL, A. R.; OLIVEIRA, F. D. Um estudo sobre classificação de erros: uma proposta aplicada a objetos de aprendizagem. In: Anais do Simpósio Brasileiro de Informática na Educação. [S.l.: s.n.], 2011. v. 1. Citado 2 vezes nas páginas 27 e 30.

LEITE, M. D.; PIMENTEL, A. R.; PIETRUCHINSKI, M. H. Remediação de erros baseada em múltiplas representações externas e classificação de erros aplicada a objetos de aprendizagem inteligentes. In: Congresso Brasileiro de Informática na Educação. [S.l.: s.n.], 2012. Citado na página 30.

LUI, A. Teaching in the zone: An introduction to working within the zone of proximal development (ZPD) to drive effective early childhood instruction. Children's progress, p. 1–10, 2012. Citado 2 vezes nas páginas 101 e 102.

LUKEY, F. Understanding and debugging programs. International Journal of Man-Machine Studies, Elsevier, v. 12, n. 2, p. 189–202, 1980. Citado na página 27.

MACHADO, S. F.; TERUYA, T. K. Mediação pedagógica em ambientes virtuais de aprendizagem: a perspectiva dos alunos. In: . [S.l.: s.n.], 2009. Citado na página 30.

MARCZAL, D. FARMA: Uma ferramenta de autoria para objetos de aprendizagem de conceitos matemáticos. Tese (Doutorado) — Universidade Federal do Paraná, 2014. Citado 6 vezes nas páginas 16, 17, 23, 26, 31 e 37.

MARCZAL, D.; DIRENE, A. FARMA: Uma ferramenta de autoria para objetos de aprendizagem de conceitos matemáticos. Anais do Simpósio Brasileiro de Informática na Educação, v. 23, 2012. Citado 2 vezes nas páginas 22 e 32.

MARCZAL, D. et al. Farma: Uma ferramenta de autoria para objetos de aprendizagem de conceitos matemáticos. In: Anais dos Workshops do Congresso Brasileiro de Informática na Educação. [S.l.: s.n.], 2015. v. 4, n. 1, p. 23. Citado 4 vezes nas páginas 15, 22, 27 e 36.

MARCZAL, D.; DIRENE, A. I. Um arcabouço que enfatiza a retroação a contextos de erro na solução de problemas. Revista Brasileira de Informática na Educação, v. 19, n. 01, p. 63, 2011. Citado 2 vezes nas páginas 16 e 32.

MARTINS, J. C. Vygotsky e o papel das interações sociais na sala de aula: reconhecer e desvendar o mundo. Série Idéias, v. 28, p. 111–122, 1997. Citado na página 26.

MARTINS, L. M. Os fundamentos psicológicos da pedagogia histórico-crítica e os fundamentos pedagógicos da psicologia histórico-cultural. Germinal: Marxismo e Educação em Debate, v. 5, n. 2, p. 130–143, 2014. Citado 2 vezes nas páginas 16 e 25.

MASCHIO, E. Modelagem do processo de aquisição de conhecimento apoiado por ambientes inteligentes. Tese (Doutorado) — Universidade Federal do Paraná, Setor de Ciências Exatas, Programa de Pós-graduação em Informática, 2013. Citado 2 vezes nas páginas 24 e 25.

MCLAREN, B. M. et al. To err is human, to explain and correct is divine: A study of interactive erroneous examples with middle school math students. In: 21st Century Learning for 21st Century Skills. [S.l.]: Springer, 2012. p. 222–235. Citado na página 27.

MORENO, R.; MAYER, R. Interactive multimodal learning environments. Educational Psychology Review, Springer, v. 19, n. 3, p. 309–326, 2007. Citado na página 15.

MOURA, V. A. B. de; PERES, L. M. Evaluation of the impact of retroaction in learning supported by an educational tool. Brazilian Journal of Computers in Education, v. 25, n. 01, p. 60, 2017. Citado 3 vezes nas páginas 32, 33 e 34.

MURRAY, T.; WOOLF, B.; MARSHALL, D. Lessons learned from authoring for inquiry learning: A tale of authoring tool evolution. In: SPRINGER. Intelligent Tutoring Systems. [S.l.], 2004. p. 197–206. Citado na página 22.

NASCIMENTO, J. K. F. d. Informática aplicada à educação. Brasília: Universidade de Brasília, 2007. Citado na página 21.

NOUSHAD, P. Cognitions about cognitions: The theory of metacognition. Online Submission, 2008. Citado na página 22.

PATTO, M. H. S. O fracasso escolar como objeto de estudo: anotações sobre as características de um discurso. Cadernos de pesquisa, v. 65, p. 72–77, 1988. Citado na página 28.

RAABE, A. L. A.; GIRAFFA, L. M. M. Uma arquitetura de tutor para promover experiências de aprendizagem mediadas. Simpósio Brasileiro de Informática na Educação, SBIE, 2006. Citado na página 31.

RAMOS, G. S. Detecção e remediação de erros na generalização de conceitos matemáticos por meio de sistemas tutores inteligentes. 2011. Citado 2 vezes nas páginas 15 e 30.

RAMOS, J. T. Uma análise comparativa entre o ensino médio regular e o ensino médio integrado/profissionalizante em computação: um estudo de caso. [S.l.]: UNEMAT - Universidade do Estado de Mato Grosso - Campus de Sinop, 2017. Citado 2 vezes nas páginas 85 e 86.

RAPKIEWICZ, C. E. et al. Estratégias pedagógicas no ensino de algoritmos e programação associadas ao uso de jogos educacionais. RENOTE, v. 4, n. 2, 2006. Citado na página 25.

SFORNI, M. S. d. F. Aprendizagem conceitual e organização do ensino: contribuições da teoria da atividade. Araraquara: JM Editora, 2004. Citado na página 28.

SHAPIRO, E. Y. Algorithmic Program Debugging, ACM Distinguished Dissertation Series. [S.l.]: MIT Press, Cambridge, MA, 1982. Citado na página 27.

SILVA, A. P. Aplicações de sistemas tutores inteligentes na educação a distância: possibilidades e limites. Anais do Seminário Nacional ABED de Educação a Distância. Brasília: ABED, 2006. Citado na página 34.

SILVA, J. R.; GERMANO, J. S.; MARIANO, R. S. Simquest-ferramenta de modelagem computacional para o ensino de Física. Revista Brasileira de Ensino de Física, v. 33, n. 1, p. 1508, 2011. Citado na página 22.

SOARES, S. M. L. de M. O sociointeracionismo: Um desafio para professor da educação básica. 2005. Citado na página 17.

THARP, R. G.; GALLIMORE, R. Rousing minds to life: Teaching, learning, and schooling in social context. [S.l.]: Cambridge University Press, 1991. Citado na página 103.



VICARI, R. M.; GIRAFFA, L. M. Fundamentos dos sistemas tutores inteligentes. Sociedades artificiais: a nova fronteira da inteligência das máquinas. Porto Alegre: Bookman. ISBN, p. 85–363, 2003. Citado na página 34.

VYGOTSKY, L. S. Mind in society: The development of higher psychological processes. [S.l.]: Harvard university press, 1980. Citado na página 26.

VYGOTSKY, L. S. Thinking and speech. the collected works of ls vygotsky, vol. 1. New York: Plenum Press, v. 114, p. 113–114, 1987. Citado na página 101.

VYGOTSKY, L. S. A formação social da mente. Psicologia, v. 153, p. V631, 1989. Citado na página 26.

VYGOTSKY, L. S. Obras escogidas tomo i. Madrid, Visor, 1997. Citado na página 16.

WEISZ, T.; SANCHEZ, A. O diálogo entre o ensino e a aprendizagem. São Paulo: Ática, v. 2, 2002. Citado na página 32.

WILEY, D. A. Connecting learning objects to instructional design theory: A definition, a metaphor, and a taxonomy. [S.l.: s.n.], 2003. Citado na página 15.

YACEF, K. Intelligent teaching assistant systems. In: IEEE. Computers in Education, 2002. Proceedings. International Conference on. [S.l.], 2002. p. 136–140. Citado na página 31.

## APÊNDICE 1 - LISTA DE QUESTÕES

Tabela 6.1: Lista das questões com enunciados

| Questão                   | Enunciado  |
|---------------------------|--|
| Conversão de temperaturas | Fazer um programa em Pascal que lê um número representando uma temperatura na escala Celsius e imprime a correspondente temperatura na escala Fahrenheit. Observação: $C = 5*(F-32)/9$ .<br>Entrada: 100<br>Saída: 212.00 graus Fahrenheit   |
| Conversão de velocidade   | Fazer um programa em Pascal que lê um número representando uma velocidade em km/h e imprime correspondente velocidade em ml/h (milhas por hora). Observação: 1 milha = 1.609344 km.<br>Entrada: 100<br>Saída: 62.14 ml/h   |
| Graus, minutos e segundos | Dado um número inteiro que representa uma quantidade de segundos (unidade de medida de ângulo geométrico), determinar o seu valor equivalente em graus, minutos e segundos. Se a quantidade de segundos for insuficiente para dar o valor 1 (um) em graus, o valor em graus deve ser 0 (zero). A mesma observação vale em relação aos minutos e segundos. Por exemplo: 3.600 segundos = 1 grau, 0 minutos, 0 segundos. Outro exemplo: 3.500 segundos = 0 graus, 58 minutos e 20 segundos.<br>Entrada: 3500<br>Saída: 0 graus, 58 minutos e 20 segundos   |
| Frações                   | Dadas duas frações ordinárias a/b e c/d, determinar a sua soma e o seu produto, também no formato de frações. A entrada de dados deve ser constituída de duas linhas, cada uma contendo dois inteiros, uma para a e b e outra para c e d. A saída deverá ser também de duas linhas cada uma contendo um par que representa o numerador e o denominador tanto da soma como do produto calculados.<br>Entrada:<br>1 2<br>3 4<br>Saída:<br>A soma = 10 / 8<br>O produto = 3 / 8   |
| Sistema de equações       | Fazer um programa em Pascal que lê seis números, a11, a12, b1, a21, a22 e b2 e imprime uma solução do sistema de equações lineares abaixo:<br>$a_{11}x + a_{12}y = b_1$<br>$a_{21}x + a_{22}y = b_2$<br>Entrada:<br>1 2 3<br>3 2 1<br>Saída:<br>x = -1.00<br>y = 2.00  |
| Fatores primos            | Fazer um programa em Pascal capaz de ler apenas um número inteiro positivo do teclado, de calcular a decomposição em fatores primos desse número e de imprimir os fatores calculados. Exemplos de entrada e saída são:<br><br>./fatores\_primos<br>12 <ENTER><br>2 2 3<br><br>./fatores\_primos<br>150 <ENTER><br>2 3 5 5  |
| Co-seno de 1 radiano      | Considere a soma S dos termo da série in nita apresentada abaixo, a qual é responsável pelo cálculo do valor do co-seno de 1 (um) radiano:<br>$S = 1 - \frac{1}{2!} + \frac{1}{4!} - \frac{1}{6!} + \frac{1}{8!} - \frac{1}{10!} + \frac{1}{12!} - \dots$<br>Fazer um programa em linguagem Pascal que seja capaz de calcular o valor aproximado da soma (S) dos termos da série até o momento em que a diferença das normas (módulo) de 2 termos consecutivos for menor que 0,000001 (i.e., norma da diferença das normas de dois termos consecutivos). Este programa não possui nenhuma entrada. |
| É palíndromo?             | Fazer um programa em Pascal capaz de ler apenas um número inteiro positivo do teclado e de verificar se a sua sequência de dígitos é exatamente a mesma, tanto se for visto da esquerda para a direita quanto da direita para a esquerda. Caso seja, imprimir “Sim eh palindromo”. Caso não seja, imprimir “Nao eh palindromo” (sem acento). Dica: aplique divisões sucessivas por 10. Exemplos de entrada e saída são:<br><br>./eh\_palindromo<br>12321 <ENTER><br>Sim eh palindromo<br><br>./eh\_palindromo<br>31527 <ENTER><br>Nao eh palindromo  |

*Continua na próxima página*

Tabela 6.1 – Continuação da página anterior

| Questão                        | Enunciado   |     |     |    |               |   |   |     |     |     |     |    |               |     |     |    |    |   |  |
|--------------------------------|---|-----|-----|----|---------------|---|---|-----|-----|-----|-----|----|---------------|-----|-----|----|----|---|--|
| É k-alternante?                | <p>Dizemos que uma sequência de inteiros é k-alternante se for composta alternadamente por segmentos de números pares de tamanho k e segmentos de números ímpares de tamanho k. Exemplos:</p> <p>A sequência 1 3 6 8 9 11 2 4 1 7 6 8 é 2-alternante.</p> <p>A sequência 2 1 4 7 8 9 é 1-alternante.</p> <p>A sequência 4 2 3 1 6 4 2 9 não é alternante.</p> <p>A sequência 1 3 5 é 3-alternante.</p> <p>Fazer um programa em Pascal capaz de ler os seguintes valores de entrada:</p> <p>a) um número inteiro k positivo que representa o tamanho do segmento alternante que será usado para testar a sequência;</p> <p>b) uma sequência de números inteiros positivos terminada por zero (o zero não faz parte da sequência). O programa deve determinar se a sequência de números lidos é k-alternante. Se for, o programa deve imprimir “Sim eh k-alternante”. Caso contrário, deve ser impressa a mensagem “Nao eh k-alternante” (sem acento). Exemplos de entrada e saída são:</p> |     |     |    |               |   |   |     |     |     |     |    |               |     |     |    |    |   |  |
|                                | <pre>./eh_k_alternante 2 &lt;ENTER&gt; 1 3 6 8 9 11 2 4 1 7 6 8 0 &lt;ENTER&gt; Sim eh k-alternante  ./eh_k_alternante 3 &lt;ENTER&gt; 4 2 8 3 1 7 6 4 6 2 9 7 0 &lt;ENTER&gt; Nao eh k-alternante</pre>  |     |     |    |               |   |   |     |     |     |     |    |               |     |     |    |    |   |  |
| Vigésimo terceiro número primo | Fazer um programa em linguagem Pascal que seja capaz de calcular e imprimir o vigésimo terceiro número primo.   |     |     |    |               |   |   |     |     |     |     |    |               |     |     |    |    |   |  |
| Multiplicação                  | <p>Fazer um programa em PASCAL para simular o processo de multiplicação ensinado por uma professora primária aos seus alunos. O programa deve funcionar para multiplicandos e multiplicadores que são números inteiros com qualquer quantidade de algarismos, mesmo que o programa inevitavelmente seja testado apenas com entradas e saídas que estão dentro da faixa de valores permitidos para variáveis do tipo Integer do Pascal padrão ou do tipo Longint do dialeto Free Pascal Compiler. Os dois números, multiplicando e multiplicador, são fornecidos por meio de leitura do teclado do computador. Veja um exemplo de entrada e saída:</p>   |     |     |    |               |   |   |     |     |     |     |    |               |     |     |    |    |   |  |
|                                | <pre>./multiplica 2187 &lt;ENTER&gt; 14 &lt;ENTER&gt; 30618</pre> <p>OBSERVAÇÃO: Aplique divisões sucessivas por 10.</p>  |     |     |    |               |   |   |     |     |     |     |    |               |     |     |    |    |   |  |
| MDC                            | <p>Fazer um programa em linguagem Pascal para ler uma quantidade arbitrária de números inteiros positivos do teclado e calcular o MDC (máximo divisor comum) de todos eles pelo método Euclidiano. Ao final, imprimir o valor do MDC calculado. O número zero é o último lido e não deve ser levado em conta no cálculo do MDC. Veja um exemplo do método Euclidiano aplicado a dois números: 720 e 520.</p>  |     |     |    |               |   |   |     |     |     |     |    |               |     |     |    |    |   |  |
|                                | <table><tr><td>***</td><td>1</td><td>2</td><td>1</td><td>1</td><td>2</td></tr><tr><td>720</td><td>520</td><td>200</td><td>120</td><td>80</td><td>40 (Resposta)</td></tr><tr><td>200</td><td>120</td><td>80</td><td>40</td><td>0</td><td></td></tr></table> <p>Veja um exemplo de entrada e saída:</p>   | *** | 1   | 2  | 1             | 1 | 2 | 720 | 520 | 200 | 120 | 80 | 40 (Resposta) | 200 | 120 | 80 | 40 | 0 |  |
| ***                            | 1   | 2   | 1   | 1  | 2             |   |   |     |     |     |     |    |               |     |     |    |    |   |  |
| 720                            | 520   | 200 | 120 | 80 | 40 (Resposta) |   |   |     |     |     |     |    |               |     |     |    |    |   |  |
| 200                            | 120   | 80  | 40  | 0  |               |   |   |     |     |     |     |    |               |     |     |    |    |   |  |
|                                | <pre>./mdc 480 120 96 4800 48 108 0 &lt;ENTER&gt; 12</pre> <p>OBSERVAÇÃO: NÃO utilize nem VETORES (ARRAYS) nem STRINGS.</p>   |     |     |    |               |   |   |     |     |     |     |    |               |     |     |    |    |   |  |
| Quadrado mágico                | <p>Um “Quadrado Mágico” de 9 (nove) elementos (3 linhas por 3 colunas) pode ser definido como a combinação dos números de 1 até 9, sem repetição, organizados de forma bidimensional tal que a soma dos elementos de quaisquer das linhas, colunas ou diagonais resulta sempre no valor 15 (quinze). Abaixo, é apresentada uma das possíveis combinações de “Quadrado Mágico”.</p>  |     |     |    |               |   |   |     |     |     |     |    |               |     |     |    |    |   |  |
|                                | <table><tr><td>6</td><td>7</td><td>2</td></tr><tr><td>1</td><td>5</td><td>9</td></tr><tr><td>8</td><td>3</td><td>4</td></tr></table> <p>Fazer um programa em linguagem PASCAL para gerar e imprimir apenas o primeiro “Quadrado Mágico” encontrado.</p> <p>OBSERVAÇÃO 1: Cuidado! Das 387.420.489 (<math>9^3</math>) variações de quadrados compostos com repetição de números de 1 a 9, somente 362.880 (<math>9!</math>) não apresentam nenhuma repetição, dos quais menos de 10 são “Quadrados Mágicos”. Mas prepare-se para esperar na frente do computador!</p> <p>OBSERVAÇÃO 2: Não utilize nem VETORES (ARRAYS) nem STRINGS.</p>   | 6   | 7   | 2  | 1             | 5 | 9 | 8   | 3   | 4   |     |    |               |     |     |    |    |   |  |
| 6                              | 7   | 2   |     |    |               |   |   |     |     |     |     |    |               |     |     |    |    |   |  |
| 1                              | 5   | 9   |     |    |               |   |   |     |     |     |     |    |               |     |     |    |    |   |  |
| 8                              | 3   | 4   |     |    |               |   |   |     |     |     |     |    |               |     |     |    |    |   |  |
| Números repetidos              | <p>Fazer um programa em linguagem Pascal para ler uma quantidade arbitrária de números inteiros positivos do teclado e determinar os números repetidos fornecidos na entrada de dados. O programa deve imprimir apenas uma ocorrência de cada número repetido, mesmo que sejam fornecidas várias duplicatas do mesmo número no momento da entrada. O número zero é o último lido e não deve ser levado em conta na determinação de repetidos. Um exemplo de entrada e saída é:</p>  |     |     |    |               |   |   |     |     |     |     |    |               |     |     |    |    |   |  |
|                                | <pre>./repetidos 3 4 5 5 6 7 8 8 9 10 5 5 5 7 7 3 0 &lt;ENTER&gt; Os repetidos sao: 5 8 7 3</pre>   |     |     |    |               |   |   |     |     |     |     |    |               |     |     |    |    |   |  |
| União                          | <p>Escreva um programa em linguagem Pascal para ler, a partir do teclado, duas sequências com quantidades arbitrárias de valores inteiros positivos, as quais denotam dois conjuntos da matemática. Os valores de uma sequência nunca conterão duplicatas mas podem vir fora de ordem. Cada sequência termina com o valor 0 (que não é elemento de nenhum dos conjuntos). Depois da leitura dos dados, o programa deve determinar e imprimir o conjunto união. Por exemplo, considere a entrada e a saída de dados abaixo:</p>  |     |     |    |               |   |   |     |     |     |     |    |               |     |     |    |    |   |  |
|                                | <pre>./uniao 11 7 1 18 6 5 9 0 &lt;ENTER&gt; 9 22 4 5 6 18 0 &lt;ENTER&gt; Conjunto Uniao = 11 7 1 18 6 5 9 22 4</pre>  |     |     |    |               |   |   |     |     |     |     |    |               |     |     |    |    |   |  |

Continua na próxima página

Tabela 6.1 – Continuação da página anterior

| Questão             | Enunciado   |
|---------------------|---|
| Triângulo de Pascal | <p>O “Triângulo de Pascal” é formado por uma composição específica de números inteiros. Cada um dos números do triângulo é formado pela soma de dois números da linha anterior: um imediatamente acima e o outro, acima e à esquerda. Fazer um programa em Pascal para ler um número inteiro positivo que representa a altura de um Triângulo de Pascal, gerar e imprimir os valores desse Triângulo no monitor de vídeo do computador. Veja uma exemplo de execução do programa:</p> <pre> ./triangulo 7 &lt;ENTER&gt; 1 1 1 1 2 1 1 3 3 1 1 4 6 4 1 1 5 10 10 5 1 1 6 15 20 15 6 1 </pre>   |
| Aniversário         | <p>Fazer um programa em linguagem Pascal capaz de calcular de forma numérica (não analítica) e imprimir a probabilidade de, em um grupo de 27 pessoas aleatoriamente escolhidas, ao menos 2 dessas fazerem aniversário no mesmo dia do ano. O programa deve conter ao menos um procedimento e uma função. O procedimento (“preenche”) deverá receber como parâmetro, por referência, um vetor de 27 posições de inteiros. Para cada posição do vetor, o procedimento deverá preenchê-la, aleatoriamente, com um valor inteiro entre 1 e 365 (OBS: utilizar a função interna do dialeto de Pascal para a geração de números aleatórios que estiver disponível em seu compilador). A função (“tem coincidência”), cujo parâmetro de entrada é um vetor com 27 posições preenchidas com números inteiros entre 1 e 365, retornará “true” se houver ao menos 2 posições do vetor com o mesmo valor e “false” no caso contrário.</p>   |
| Intercala           | <p>Escreva um programa completo na linguagem Pascal capaz de realizar a intercalação de duas listas de valores reais lidos do teclado. O programa deve conter um procedimento chamado intercala. Seus parâmetros são três vetores (e seus tamanhos): Va, Vb e Vc. Todos são vetores de valores Reais, sendo que Va e Vb possuem uma quantidade arbitrária de valores lidos do teclado (o zero é o último valor de cada lista de entrada). O procedimento deve intercalar os valores de Va com os de Vb e colocar o resultado em Vc. A intercalação é uma operação de montagem de uma lista com elementos que foram escolhidos alternadamente de duas outras listas. Por exemplo, suponha que os valores do vetor Va recebidos como parâmetro sejam os seguintes:</p> <pre>1.0 2.0 3.0 4.0 5.0 6.0 7.0 0 &lt;ENTER&gt;</pre> <p>Suponha também que os valores do vetor Vb sejam os seguintes:</p> <pre>11.0 12.0 13.0 8.0 0 &lt;ENTER&gt;</pre> <p>O procedimento intercala deve retornar os seguintes valores por meio do vetor Vc a quem o ativar:</p> <pre>1.0 11.0 2.0 12.0 3.0 13.0 4.0 8.0 5.0 6.0 7.0</pre> <p>Atenção: O procedimento intercala deve ser projetado para imprimir nenhum valor na saída padrão. Realize a impressão por meio de outro módulo.</p> |
| Ocorrências         | <p>Escreva um programa completo na linguagem Pascal para ler uma sequência de inteiros positivos e imprimir quantas vezes cada valor aparece na sequência. A sequência contém no máximo 100 inteiros e termina quando o valor 0 é digitado. Veja um exemplo de execução:</p> <pre> 104 8 9 12 8 73 104 9 8 1001 0 &lt;ENTER&gt; A sequencia contem: 104: 2 ocorrencias 8: 3 ocorrencias 9: 2 ocorrencias 12: 1 ocorrencia 73: 1 ocorrencia 1001: 1 ocorrencia </pre> <p>Sugestão: escreva a função abstrata busca( vet, tam, num ) que retorna o índice onde se encontra o valor num no vetor vet de tam elementos. Caso num não exista, a função retorna um código de erro (por exemplo, -1).</p>  |

*Continua na próxima página*

Tabela 6.1 – Continuação da página anterior

| Questão               | Enunciado   |
|-----------------------|---|
| Gera imagem           | <p>Fazer um programa em Pascal para preencher uma matriz de elementos inteiros com m colunas por n linhas. O valor de cada célula poderá ser um número que vai de 0 (zero) até max, onde max será definido como uma constante inteira do programa em Pascal, assim como m e n também o serão. Na primeira versão do programa, m deverá valer 210, n deverá valer 140 e max deverá ser 29399, assim:</p> <pre>... const m = 210; n = 140; max = 29399; ...</pre> <p>O preenchimento dos elementos da matriz durante a leitura deverá ser feito por linha (não por coluna), utilizando apenas um dos modos relacionados abaixo para a entrega da lista (o aluno poderá tentar os outros por conta própria pois são todos bem simples):</p> <ul style="list-style-type: none"> <li>• preenchimento serial (0, 1, 2, 3, . . . , 29399);</li> <li>• preenchimento aleatório (função random para valores entre 0 e 29399);</li> <li>• alguma função cíclica, cujos valores nunca ultrapassem 29399</li> </ul> <p>(exemplo: <math>29399 * \sin(x)</math>, com x variando de 0 a 29399, mas use as funções trunc e abs do Pascal para transformar a saída da função seno em número inteiro e ainda transformar números negativos em positivos);</p> <ul style="list-style-type: none"> <li>• outras possibilidades interessantes de sua própria escolha.</li> </ul> <p>O formato da impressão do programa deverá ser exatamente o mesmo de uma imagem codificada no padrão PGM (Portable Gray Map), o qual terá m + 3 linhas. A primeira linha só deverá possuir dois caracteres: P2. A segunda linha deverá ter dois inteiros: a quantidade de colunas (n) da matriz seguido da quantidade de linhas (m). A terceira linha possui apenas o valor da constante max, a qual representa o valor da máxima intensidade de luz em uma análoga escala de cinza cujos valores variam de 0 a max. Veja um exemplo de impressão se o programa tivesse n = 24 colunas, m = 7 linhas e max = 15.</p> <pre>./geraimagem   less P2 24 7 15 0 3 3 3 3 0 0 7 7 7 7 0 0 11 11 11 11 0 0 15 15 15 15 0 0 3 0 0 0 0 0 7 0 0 0 0 0 11 0 0 0 0 0 15 0 0 15 0 0 3 3 3 0 0 0 7 7 7 0 0 0 11 11 11 0 0 0 15 15 15 15 0 0 3 0 0 0 0 0 7 0 0 0 0 0 11 0 0 0 0 0 15 0 0 0 0 0 3 0 0 0 0 0 7 7 7 7 0 0 11 11 11 11 0 0 15 0</pre> <p>Apenas a título de curiosidade, é possível visualizar a imagem PGM gerada pelo programa usando o utilitário display, assim:</p> <pre>./geraimagem   display &amp;</pre> <p>Obs.: O Farma-Alg não irá testar as respostas para este exercício com nenhum caso de teste. Todas as respostas serão consideradas corretas, até que o tutor as corrija manualmente.</p> |
| Matriz de Vandermonde | <p>Fazer um programa em Pascal para ler dois números inteiros que representam, respectivamente, a quantidade m de linhas e n de colunas de uma matriz. Em seguida, devem ser lidos os m x n elementos da matriz, os quais são números reais, entrados pela ordem das linhas da matriz. Depois da leitura, o programa deve decidir e imprimir se ela é ou não uma Matriz de Vandermonde. Se for, imprimir Sim. Se não for, imprimir Nao.</p> <p>Pesquise sobre Matriz de Vandermonde em:</p> <p><a href="http://pt.wikipedia.org/wiki/Matriz_de_Vandermonde">http://pt.wikipedia.org/wiki/Matriz_de_Vandermonde</a></p> <p>De forma resumida, uma Matriz de Vandermonde é aquela que segue a forma geral abaixo:</p> $V = \begin{bmatrix} 1 & \alpha_1 & \alpha_1^2 & \dots & \alpha_1^{n-1} \\ 1 & \alpha_2 & \alpha_2^2 & \dots & \alpha_2^{n-1} \\ 1 & \alpha_3 & \alpha_3^2 & \dots & \alpha_3^{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha_m & \alpha_m^2 & \dots & \alpha_m^{n-1} \end{bmatrix}$ <p>Veja um exemplo de execução do programa abaixo:</p> <pre>5 4 &lt;ENTER&gt; 1 2 4 16 &lt;ENTER&gt; 1 3 9 81 &lt;ENTER&gt; 1 4 16 256 &lt;ENTER&gt; 1 5 25 625 &lt;ENTER&gt; 1 6 36 1296 &lt;ENTER&gt; Nao</pre> <p>Veja outro exemplo:</p> <pre>5 4 &lt;ENTER&gt; 1 2 4 8 &lt;ENTER&gt; 1 3 9 27 &lt;ENTER&gt; 1 4 16 64 &lt;ENTER&gt; 1 5 25 125 &lt;ENTER&gt; 1 6 36 216 &lt;ENTER&gt; Sim</pre>   |

Continua na próxima página

Tabela 6.1 – Continuação da página anterior

| Questão                | Enunciado   |
|------------------------|---|
| Deslocamento de imagem | <p>Esta lista possui um único exercício, o qual foi projetado para ser resolvido durante o período de uma semana. Ele adiciona características ao Exercício 3 da Lista 5 e, por isso, sua solução pode ser facilitada se for uma modificação da solução anteriormente elaborada:</p> <p><a href="http://www.inf.ufpr.br/alexander/ci055/lista_5.pdf">http://www.inf.ufpr.br/alexander/ci055/lista_5.pdf</a></p> <p>Tais características adicionais são muito semelhantes ao que consta no enunciado de problema no link abaixo, porém apenas um deslocamento horizontal e um vertical será permitido agora:</p> <p><a href="http://br.spoj.com/problems/TV/">http://br.spoj.com/problems/TV/</a></p> <p>Fazer um programa em Pascal para ler, da entrada padrão (i.e., teclado), os dados de uma imagem de formato padrão PGM (Portable Gray Map). A leitura dos elementos da matriz deverá ser feita linha-a-linha (não por coluna).</p> <p>O formato da impressão do programa, sempre na saída padrão (i.e., monitor de vídeo), deverá ser exatamente o mesmo de uma imagem codificada no padrão PGM (Portable Gray Map), o qual terá <math>m+3</math> linhas. A primeira linha só deverá possuir dois caracteres: P2. A segunda linha deverá ter dois inteiros: a quantidade de colunas da matriz seguido da quantidade de linhas. A terceira linha possui apenas o valor da constante max, a qual representa o valor da máxima intensidade de luz em uma análoga escala de cinza cujos valores variam de 0 a max. Veja um exemplo de arquivo PGM com <math>n = 4</math> colunas, <math>m = 7</math> linhas e <math>\text{max} = 12</math>.</p> <pre>cat texto\imagem\1.pgm P2 4 3 12 6 7 8 5 10 11 12 9 2 3 4 1</pre> <p>Os deslocamentos horizontal e vertical da imagem devem ser lidos também da entrada padrão por meio de dois números inteiros. O primeiro valor representa o deslocamento na direção horizontal (valor positivo representa deslocamento para a direita e valor negativo para a esquerda). O segundo valor representa o deslocamento da direção vertical (valor positivo para cima e valor negativo para baixo). Esses dois valores constituem o final da leitura de dados do programa. Veja um exemplo de arquivo com dados de deslocamento horizontal de 3 pontos para a esquerda e vertical de 2 pontos para cima.</p> |

*Continua na próxima página*

Tabela 6.1 – Continuação da página anterior

| Questão                        | Enunciado   |
|--------------------------------|---|
| Deslocamento de imagem (cont.) | <pre>cat desloc_h_-3_v_2.txt -3 2</pre> <p>Após a leitura, o programa deve efetuar ambos os deslocamentos sobre a matriz da imagem e, em seguida, imprimir a imagem completa na saída padrão no formato PGM. Por exemplo, se considerarmos os dois arquivos acima, o programa deverá imprimir os seguintes dados na tela.</p> <pre>cat texto_imagem_1.pgm desloc_h_-3_v_2.txt   ./sol_q1_lista6   less P2 4 3 12 1 2 3 4 5 6 7 8 9 10 11 12</pre> <p>Tal como já foi explicado anteriormente, também é possível visualizar a imagem PGM gerada pelo programa usando o utilitário display. Tente copiar a imagem PGM que está no link:</p> <p><a href="http://www.inf.ufpr.br/cursos/ci055/imagem_Algoritmos.pgm">http://www.inf.ufpr.br/cursos/ci055/imagem_Algoritmos.pgm</a></p> <p>Execute o utilitário display para mostrar a imagem graficamente, assim:</p> <pre>display imagem_Algoritmos.pgm &amp;</pre> <p>Veja na Figura 1 como a imagem aparece na sua tela:</p>  <p>Figura 1: Imagem sem deslocamento apresentada com o utilitário display.</p> <p>Crie também um exemplo de arquivo com dados de deslocamento horizontal de 48 pontos para a direita e vertical de 23 pontos para baixo. Veja como ele fica ao ser listado:</p> <pre>cat desloc_h_48_v_-23.txt 48 -23</pre> <p>Agora, provoque a entrada de dados por meio do utilitário cat aplicado, na ordem de entrada, aos dois arquivos. Canalice a saída do cat como entrada para o programa construído e, em seguida, canalice a saída para a execução do utilitário display por último para mostrar a imagem, assim:</p> <pre>cat imagem_Algoritmos.pgm desloc_h_48_v_-23.txt   ./sol_q1_lista6   display &amp;</pre> <p>Veja na Figura 2 como a imagem aparece na sua tela:</p>  <p>Figura 2: Imagem sem deslocamento apresentada com o utilitário display.</p> <p>Os arquivos citados nos exemplos acima podem ser obtidos nos links abaixo:</p> <p><a href="http://www.inf.ufpr.br/cursos/ci055/imagem_Algoritmos.pgm">http://www.inf.ufpr.br/cursos/ci055/imagem_Algoritmos.pgm</a><br/> <a href="http://www.inf.ufpr.br/cursos/ci055/desloc_h_48_v_-23.txt">http://www.inf.ufpr.br/cursos/ci055/desloc_h_48_v_-23.txt</a><br/> <a href="http://www.inf.ufpr.br/cursos/ci055/texto_imagem_1.pgm">http://www.inf.ufpr.br/cursos/ci055/texto_imagem_1.pgm</a><br/> <a href="http://www.inf.ufpr.br/cursos/ci055/desloc_h_-3_v_2.txt">http://www.inf.ufpr.br/cursos/ci055/desloc_h_-3_v_2.txt</a></p> <p>Uma possível composição do corpo de comandos do Programa Principal poderia ser o seguinte (note que a execução dos dois comandos write foi inibida por meio de comentário de código):</p> <pre>[language=Pascal] begin le_matriz(mat, m, n, max); (* write('Entre com o deslocamento Horizontal da imagem (negativo se for para a esquerda): '); *) read(deltah); (* write('Entre com o deslocamento Vertical da imagem (negativo se for para baixo): '); *) read(deltav); desloca_matriz(mat, m, n, deltah, deltav); imprime_matriz(mat, m, n, max); end.</pre> <p>Finalmente, como sugestão, tente usar o software gimp para criar imagens e salvar em formato PGM para testar seu programa. Atenção ao salvar o arquivo para que a opção ASCII seja adotada (não utilize a opção raw).</p> |

Continua na próxima página

Tabela 6.1 – Continuação da página anterior


| Questão | Enunciado   |
|---------|---|
| Sinuca  | <p>Esta lista possui um único exercício, o qual foi projetado para ser resolvido durante o período de uma semana. Ele adiciona características aos exercícios já resolvidos em sala de aula que adotam reduções do tamanho da memória da estrutura de dados quando o objeto principal é uma matriz esparsa. Veja o exemplo de código que decide se nenhuma das <math>n</math> Torres de um tabuleiro de xadrez de <math>n</math> por <math>n</math> ataca a outra. Nesse programa fonte, o tabuleiro que tem <math>n \times n</math> casas é uma matriz esparsa, a qual foi representada na forma de um vetor de inteiros onde a dezena de cada inteiro guarda a linha onde há uma Torre e a unidade guarda a coluna.</p> <p><a href="http://www.inf.ufpr.br/cursos/ci055/ntorres.pas">http://www.inf.ufpr.br/cursos/ci055/ntorres.pas</a></p> <p>Agora, neste exercício, além de lidar com o mesmo conceito de matrizes esparsas, os primeiros passos de simulação computacional serão dados por meio da animação aproximada do movimento da bola branca em uma mesa de sinuca. Para tanto, vamos precisar de alguns conceitos da Geometria, da Trigonometria e, claro, da Física. Todavia, atenção: nesta simulação, a noção de tempo não é fundamentalmente o da Física mas, sim, o tempo discreto de um ciclo de re-cálculo da posição da bola branca. Isso tem implicações sobre o retrato da mesa em cada um dos quadros da simulação, os quais não são função do tempo Físico.</p> <p>Adicionalmente, neste mundo ideal, não há colisão entre bolas na primeira versão do programa. Além disso, apenas a bola branca se movimenta e colide contra as laterais da mesa, seguindo então uma trajetória de um ângulo reflexivo de acordo com as leis de Física.</p> <p>Fazer um programa em Pascal para ler, da entrada padrão (i.e., teclado), os dados de dimensões de uma mesa de sinuca, a quantidade de bolas (incluindo a bola branca) e a localização linha e coluna de cada bola (iniciando com a da bola branca).</p> <p>O formato da impressão do programa, sempre na saída padrão (i.e., monitor de vídeo), deverá ser semelhante ao de uma mesa de sinuca disposta com a Largura na orientação vertical e o Comprimento na horizontal. Incluindo as bordas, tanto para a leitura como para a impressão, a Largura (em Linhas) e o Comprimento (em Colunas) são limitadas a 40 unidades da tela (chamadas aqui de pixels). A máxima quantidade de bolas é de 9.</p> <p>Além desses dados, mais dois são lidos ainda: a norma do vetor velocidade inicial (em pixels por segundo) e o ângulo de movimento da bola branca (em graus). As medidas de ângulo são de valores reais maiores ou iguais a 0 (zero) ou menores do que 360. A orientação positiva do ângulo é no sentido anti horário (da Trigonometria) e com seu vértice de medição centrado na bola.</p>  <p>Veja uma forma aproximada da saída padrão para o exemplo acima, em que a mesa de sinuca tem Largura = 10 pixels, Comprimento = 40 pixels e a quantidade de 7 bolas.</p> <p>Após a leitura, o programa deve dar início ao procedimento de simulação. Tente executar uma solução já implementada para ver um comportamento aproximado do seu programa. Primeiro, baixe os seguintes arquivos para o seu diretório de trabalho:</p> <p><a href="http://www.inf.ufpr.br/cursos/ci055/mesasinuca">http://www.inf.ufpr.br/cursos/ci055/mesasinuca</a><br/> <a href="http://www.inf.ufpr.br/cursos/ci055/dadossinuca.txt">http://www.inf.ufpr.br/cursos/ci055/dadossinuca.txt</a></p> <p>O arquivo <code>mesasinuca</code> não é um programa fonte mas apenas um código executável em Linux. Por isso, mude a permissão do arquivo <code>mesasinuca</code> para ele ficar executável e, em seguida, basta executar assim:</p> <pre>chmod u+rx mesasinuca ./mesasinuca &lt; dadossinuca.txt</pre> <p>O arquivo <b><code>dadossinuca.txt</code></b> tem todos os dados de entrada. Para o exemplo anterior, veja que ele contém:</p> <pre>cat dadossinuca.txt 10 40 7 5 7 3 30 4 31 5 32 6 33 7 31 8 27 2 325</pre> <p>Se o programa <b><code>mesasinuca</code></b> for executado sem a entrada do arquivo, o aspecto da saída/entrada padrão será:</p> <pre>./mesasinuca Entre com a Largura da mesa em pixels (no maximo 40): 10 Entre com o Comprimento da mesa em pixels (no maximo 40): 40 Entre com a quantidade de Bolas da mesa (no maximo 9): 7 Entre com a posicao da Largura da Bola Branca: 5 Entre com a posicao do Comprimento da Bola Branca: 7 Entre com a posicao da Largura da Bola 2: 3 Entre com a posicao do Comprimento da Bola 2: 30 Entre com a posicao da Largura da Bola 3: 4 Entre com a posicao do Comprimento da Bola 3: 31 Entre com a posicao da Largura da Bola 4: 5 Entre com a posicao do Comprimento da Bola 4: 32 Entre com a posicao da Largura da Bola 5: 6 Entre com a posicao do Comprimento da Bola 5: 33 Entre com a posicao da Largura da Bola 6: 7 Entre com a posicao do Comprimento da Bola 6: 31 Entre com a posicao da Largura da Bola 7: 8 Entre com a posicao do Comprimento da Bola 7: 27 Entre com a Velocidade Inicial da Bola Branca em pixels/s (de 0.01 a 3.0): 2 Entre com o Angulo Inicial da Bola Branca em graus (de 0 a 359): 325</pre> <p>Crie outros exemplos de entrada para inspecionar o comportamento aproximado do programa. Nesta primeira versão, não se preocupe tanto com as leis da Física para dar realismo à queda de velocidade da bola branca ao longo do tempo. Qualquer tipo de frenagem será boa. Para facilitar a sua solução, use os seguintes recursos:</p> <p>Os procedimentos de leitura e impressão aproximados do problema das <math>n</math>-Torres;<br/> As funções trigonométricas <math>\sin</math> (seno) e <math>\cos</math> (coseno), mas cuidado pois os argumentos precisam ser passados em radianos quando a entrada de dados deste programa deve ser fornecida em graus;<br/> Use a biblioteca do <code>fpc</code> chamada de <code>crt</code> pois com ela você poderá usar as funções <code>clrscr</code> (limpa a tela), <code>delay</code> (que retarda o processamento em um certo tempo oassado em milisegundos).</p> |



Tabela 6.1 – Continuação da página anterior

| Questão        | Enunciado   |
|----------------|---|
| Sinuca (cont.) | <p>Uma possível composição do corpo de comandos do Programa Principal poderia ser o seguinte:</p> <pre> [language=Pascal] program mesa_de_sinuca; uses crt; const   maxbolas = 9;   PI = 3.14159265; type   vetor = array[1..maxbolas] of real; var   L, C, B: integer;   Vel, Ang: real;   mesa: vetor;  ... ...  begin   write('Entre com a Largura da mesa em pixels (no maximo 40): ');   read(L);   write('Entre com o Comprimento da mesa em pixels (no maximo 40): ');   read(C);   write('Entre com a quantidade de Bolas da mesa (no maximo 9): ');   read(B);   le_posicao_das_bolas(mesa, L, C, B);   write('Entre com a Velocidade Inicial da Bola Branca em pixels/s (de 0.01 a 3.0): ');   read(Vel);   write('Entre com o Angulo Inicial da Bola Branca em graus (de 0 a 359): ');   read(Ang);   clrscr;   imprime_mesa(mesa, L, C, B);   simula(mesa, L, C, B, Vel, Ang); end. </pre> <p>Finalmente, se houver tempo, como sugestão, tente inventar uma forma de colisão entre bolas para que mais de uma delas esteja em movimento em um dado instante. Talvez seja bom usar a função random para gerar algum tipo de impressão de dispersão das bolas nas colisões. Tente também alterar a representação da matriz esparsa para receber mais do que 9 bolas na mesa.</p> |

## APÊNDICE 2 - PROCEDIMENTOS NO MONGODB

A importação do *dump* foi efetuada para a nova base com o comando

```
mongorestore --drop --db farma-alg farma_production
```

em que *farma-alg* é nome da base criada e *farma\_production* é o diretório em que se encontram os arquivos *.bson* e *.json*, fornecidos pelo autor e mantenedor da FARMA-ALG.

Foi utilizado o script Python 6.1, disponível em<sup>1</sup> para os procedimentos de exportação para CSV.

Para cada coleção do MongoDB, o procedimento para exportação seguiu conforme a seguinte sintaxe:

```
python3 mongoexportcsv.py <base> <coleção> <arquivo.csv>
```

Código 6.1: Script de exportação de MongoDB para CSV

```

1  import unicodcsv
2  import sys
3  from pymongo import MongoClient

4  class generic_converter:

5      def __init__(self):
6          self.header_dict = {}

7      def retrieve_headers(self, test_dict, name_var):
8          for element in test_dict:
9              if isinstance(test_dict[element], dict):
10                 self.retrieve_headers(test_dict[element], name_var +
11                                     '||' + element)
12             else:
13                 self.header_dict[name_var + '||' + element] =
↪ test_dict[element]

14     def converter_main(self, csv_writer):
15         mongo_uri_or_db_name = sys.argv[1]
16         if mongo_uri_or_db_name.startswith("mongodb://"): # mongodb uri given
17             client = MongoClient(mongo_uri_or_db_name)
18             db = client[mongo_uri_or_db_name.split("/")[-1]]
19         else: # database name given
20             client = MongoClient()
21             db = client[mongo_uri_or_db_name]
22             collection_obj = db[sys.argv[2]]
23             cursor_records = collection_obj.find()
24             header_list = []

```

<sup>1</sup> <<https://github.com/surya-shodan/mongoexportcsv>>

```

25     for cursor in cursor_records:
26         self.retrieve_headers(cursor, '')
27         for item_label in self.header_dict:
28             if item_label not in header_list:
29                 header_list.append(item_label)
30         self.header_dict = {}
31     csv_writer.writerow(header_list)

32     cursor_records = collection_obj.find()
33     for cursor in cursor_records:
34         row_to_push = []
35         self.header_dict = {}
36         self.retrieve_headers(cursor, '')
37         for item_label in header_list:
38             if item_label in self.header_dict:
39                 row_to_push.append(self.header_dict[item_label])
40             else:
41                 row_to_push.append('')
42     csv_writer.writerow(row_to_push)

43 def main():
44     f_write = open(sys.argv[3], 'wb')
45     csv_writer = unicodedcsv.writer(f_write, delimiter=',', quotechar='"')
46     converter_object = generic_converter()
47     converter_object.converter_main(csv_writer)

48 if __name__ == '__main__':
49     main()

```

## APÊNDICE 3 - PROCEDIMENTOS NO MYSQL

A importação dos arquivos CSV para a nova base MySQL foi efetuada pela ferramenta gráfica web PHPMYAdmin<sup>2</sup>. As consultas efetuadas foram feitas usando principalmente o recurso de SELECT combinado com LEFT JOIN, conforme Código 6.2.

Código 6.2: Consultas no MySQL

```

1 #####
2 ## Renomeando campos ##
3 #####
4 update tags
5 set name = replace(name, "Tempo", "012")
6
7 #####
8 ## Busca por respostas que receberam intervencao ##
9 #####
10
11 SELECT
12 u.sign_in_count AS Qtd_login,
13 u.name AS Aluno,
14 te.name AS Turma,
15 a.created_at AS Enviou_em,
16 a.try_number AS Tentativa,
17 a.response AS Resposta,
18 a.updated_at AS Atualizada_em,
19 sec_to_time( timestampdiff( SECOND , a.created_at, a.updated_at ) ) AS Tempo,
20 q.title AS Questao,
21 a.correct AS Correta,
22 t.description AS Erro,
23 c.name AS Professor,
24 c.text AS Comentou,
25 c.created_at AS Data_Comentario,
26 sec_to_time( timestampdiff(SECOND , a.created_at, c.created_at ) ) AS
    ↳ Tempo_apos
27 FROM answers a
28 LEFT JOIN users u ON u._id = a.user_id
29 LEFT JOIN teams te ON te._id = a.team_id
30 LEFT JOIN questions q ON q._id = a.question_id
31 LEFT JOIN tags t ON t._id = a.tag_ids
32 LEFT JOIN (SELECT c2.*, u2.name FROM comments c2 LEFT JOIN users u2 ON u2._id
    ↳ = c2.user_id) c ON c.answer_id = a._id
33 WHERE c.text IS NOT NULL
34 ORDER BY c.text desc
35 INTO OUTFILE '/Users/ruiogawa/mestrado/respostas_intervencao.csv'
    FIELDS TERMINATED BY ','
    ENCLOSED BY '"'

```

<sup>2</sup> <<https://www.phpmyadmin.net>>

```

36 #####
37 ## Busca por respostas erradas de determinada questao ##
38 #####

39 SELECT
40 u.sign_in_count AS Qtd_login,
41 u.name AS Aluno,
42 te.name AS Turma,
43 a.created_at AS Enviou_em,
44 a.try_number AS Tentativa,
45 a.response AS Resposta,
46 a.updated_at AS Atualizada_em,
47 sec_to_time( timestampdiff( SECOND , a.created_at, a.updated_at ) ) AS Tempo,
48 q.title AS Questao,
49 a.correct AS Correta,
50 t.description AS Erro,
51 c.name AS Professor,
52 c.text AS Comentou,
53 c.created_at AS Data_Comentario,
54 sec_to_time( timestampdiff(SECOND , a.created_at, c.created_at ) ) AS
    ↳ Tempo_apos
55 FROM answers a
56 LEFT JOIN users u ON u._id = a.user_id
57 LEFT JOIN teams te ON te._id = a.team_id
58 LEFT JOIN questions q ON q._id = a.question_id
59 LEFT JOIN tags t ON t._id = a.tag_ids
60 LEFT JOIN (SELECT c2.*, u2.name FROM comments c2 LEFT JOIN users u2 ON u2._id
    ↳ = c2.user_id) c ON c.answer_id = a._id
61 WHERE te.name = "Alg - 2016/1 - ci055-D" and q.title = "Graus, minutos e
    ↳ segundos" and a.correct = "false"
62 ORDER BY u.name asc
63 INTO OUTFILE '/Users/ruiogawa/mestrado/graus_minutos_segundos.csv'
64 FIELDS TERMINATED BY ','
65 ENCLOSED BY '"'

```

## APÊNDICE 4 - PROCEDIMENTOS NO R-STUDIO

Código 6.3: Rotina do R-Studio

```

1 #####
2 ## Rotina para os testes da dissertação ##
3 #####

4 ## Lê os dados do arquivo "dados.txt" usando a primeira linha como cabeçalho
5 ## e vírgula como separador de valores
6 dados=read.table("/Users/ruiogawa/mestrado/estatisticas/dados.txt", h= T,
  ↪ dec=",")

7 ## Mostra os dados
8 dados

9 # Carrega pacotes necessários
10 require(agricolae)
11 require(lattice)
12 require(MASS)
13 require(multcomp)
14 require(multcompView)
15 require(urca)
16 require(vars)
17 require(dse)
18 require(tseries)
19 require(seasonal)
20 require(tidy)
21 require(dplyr)
22 require(showtext)
23 require(ggplot2)
24 require(scales)
25 require(gplots)

26 #####
27 ## variável tent_1_int contra tent_ok ##
28 #####

29 ## Acessa os dados
30 attach(dados)
31 ## Cria o dataframe "analise" somente com as tentativas
32 analise=cbind(var_1,tentativa)
33 ## Calcula as médias das tentativas e as armazena em "media"
34 media=tapply(var_1,tentativa,mean)
35 ## Calcula o desvio padrão das tentativas e os armazena em "dp"
36 dp=tapply(var_1,tentativa,sd)
37 ## Calcula o coeficiente de variação das tentativas e os armazena em "cv"
38 cv=(dp/media)*100
39 ## Cria o dataframe "descritiva" com os resultados da média, desvio padrão e
  ↪ coeficiente de variância

```

```

40 descritiva=rbind(media,dp,cv)
41 ## Mostra os dados armazenados
42 descritiva

43 ## Construindo os Gráficos um sobre o outro ##

44 ## Armazena em "x" as informações da linha 1 ate a linha 46
45 x=dados[1:46,]
46 ## Armazena em "y" as informações da linha 47 ate a linha 92
47 y=dados[47:92,]
48 ## Carrega pacotes necessários
49 require(urca)
50 require(tseries)
51 require(vars)
52 require(dse)
53 attach(dados)
54 ## Isola os dados da coluna "var_1" (tent_1_int), da linha 1 ate a linha 46 e
   ↪ os armazena na variável "xx"
55 xx=ts(x$var_1,start=c(1,1),frequency=1)
56 ## Isola os dados da coluna "var_1" (tent_int_ok), da linha 47 ate a linha 92
   ↪ e os armazena na variável "yy"
57 yy=ts(y$var_1,start=c(1,1),frequency=1)
58 ## Permite combinar mais de um plot no mesmo gráfico, um sobre o outro
59 par(mfrow = c(1,1),family="serif",las=1,cex=1,cex.lab=1)
60 ## Plota tent_1_int ao longo dos 46 momentos
61 plot(xx, xlab="Ao longo dos 46
   ↪ momentos",ylab="Tentativas",xlim=c(1,46),ylim=c(0,37),col=1,lty=1)
62 ## Sobrepe uma linha tracejada com tent_int_ok
63 lines(yy,col="1",lty=2)
64 ## Insire um grid com espaçamento de 0,5
65 grid(col="darkgrey",lwd=0.5)
66 ## Insere a legenda no canto superior direito
67 key.space=legend("topright",cex=1,ncol=1,legend=c
68 ("tent_1_int","tent_int_ok"),lty=c(1,2),col=c(1,1))

69 ## Boxplot/Intervalo de Confiança ##

70 ## Permite combinar mais de um plot no mesmo gráfico, lado a lado
71 par(mfrow = c(1,2),family="serif",las=1,cex=1)
72 ## Cria 2 boxplots comparativos entre tent_1_int e tent_int_ok
73 boxplot(var_1~tent,xlab="Distribuição",ylab="Tentativas",pch=16)
74 ## Cria 2 plotmeans comparativos entre tent_1_int e tent_int_ok
75 plotmeans(var_1~tent,xlab="Intervalo
   ↪ confiança",ylab="Tentativas",p=0.99,barcol="black",connect=TRUE,pch=16)

76 ## Histogramas ##

77 hist(xx)
78 hist(yy)

79 ## Testes de Normalidade ##

```

```

80  ## Executa testes de Shapiro para verificar se os dados possuem distribuição
    ↪ normal
81  ## Teste de Shapiro para tent_1_int
82  shapiro.test(xx)
83  ## Teste de Shapiro para tent_int_ok
84  shapiro.test(yy)

85  ## Não houve normalidade das variáveis xx=tent_1_int e yy=tent_int_ok

86  ## Teste F para verificar a homogeneidade das variâncias entre tent_1_int e
    ↪ tent_int_ok
87  var.test(xx,yy)

88  ## Não houve homogeneidade das variâncias entre tent_1_int e tent_int_ok ##
89  ## Transformação Box Cox nos dados de tentativas ##

90  ## Carrega pacote necessário
91  require(MASS)
92  boxcox(n~factor(var_1),data=dados, plotit=T)
93  boxcox(n~factor(var_1),data=dados,lam=seq(0.70, 0.75,1/1000))
94  lambda=0.73

95  ## Transformando os Dados/Box Cox ##

96  var_1_t<- ((var_1)^(0.73) - 1)/0.73
97  par(mfrow=c(1,2))
98  xxt=var_1_t[1:46]
99  yyt=var_1_t[47:92]
100 hist(xxt)
101 hist(yyt)

102 ## Retomando os testes após transformação Box Cox ##

103 ## Testes de Normalidade ##

104 shapiro.test(xxt)
105 shapiro.test(yyt)

106 ## Não houve normalidade das variáveis xx=tent_1_int e yy=tent_int_ok ##

107 ## Teste F:homogeneidade das variâncias ##

108 var.test(xxt,yyt)

109 ## Retoma-se os dados originais e executa o teste não paramétrico de Wilcoxon
    ↪ ##

110 wilcox.test(xx,yy)

111 #####

```



```

112  ## Há diferenças entre as médias ##
113  #####

114  #####
115  ## Variável tempo_int contra tempo_int_ok ##
116  #####

117  attach(dados)
118  analise=cbind(var_2,tempo)
119  media=tapply(var_2,tempo,mean)
120  dp=tapply(var_2,tempo,sd)
121  cv=(dp/media)*100
122  descritiva=rbind(media,dp,cv)
123  descritiva

124  ## Construindo os Gráficos um sobre o outro ##

125  x=dados[1:46,]
126  y=dados[47:92,]
127  require(urca)
128  require(tseries)
129  require(vars)
130  require(dse)
131  attach(dados)
132  xx=ts(x$var_2,start=c(1,1),frequency=1)
133  yy=ts(y$var_2,start=c(1,1),frequency=1)
134  par(mfrow = c(1,1),family="serif",las=1,cex=1)
135  plot(xx,xlab="Ao longo dos 46
    ↪ momentos",ylab="Tempo",xlim=c(1,46),ylim=c(0,2119),col=1,lty=1)
136  lines(yy,col="1",lty=2)
137  grid(col="darkgrey",lwd=0.5)
138  key.space=legend("topleft",cex=1,ncol=1,legend=c
139  ("tempo_1_int","tempo_int_ok"),lty=c(1,2),col=c(1,1))

140  ## Boxplot/Intervalo de Confiança ##

141  par(mfrow = c(1,2),family="serif",las=1,cex=1)
142  boxplot(var_2~tempo,xlab="Distribuições",ylab="Tempos",pch=16)
143  plotmeans(var_2~tempo,xlab="Intervalo de
    ↪ confiança",ylab="Tempos",p=0.99,barcol="black",connect=TRUE,pch=16)

144  ## Histogramas ##

145  par(mfrow = c(1,2),family="serif",las=1,cex=0.8)
146  hist(xx)
147  hist(yy)

148  ## Testes de Normalidade ##

149  shapiro.test(xx)
150  shapiro.test(yy)

```

```

151  ## Não houve normalidade das variáveis xx=tempo_int e yy=tempo_int_ok ##
152  ## Teste F:homogeneidade das variâncias ##
153  var.test(xx,yy)
154  ## Não houve homogeneidade das variâncias ##
155  ## Transformação boxcox nos dados de tentativas ##
156  ## Não houve homogeneidade das variâncias ##
157  ## Transformação boxcox nos dados de tentativas ##
158  require(MASS)
159  boxcox(n~factor(var_2),data=dados, plotit=T)
160  boxcox(n~factor(var_2),data=dados,lam=seq(1.48, 1.49,1/1000))
161  lambda=1.488
162  ## Transformando os Dados/Box Cox ##
163  var_1_t<- ((var_2)^(1.488) - 1)/1.488
164  par(mfrow=c(1,2))
165  xxt=var_1_t[1:46]
166  yyt=var_1_t[47:92]
167  par(mfrow = c(1,2),family="serif",las=1,cex=0.8)
168  hist(xxt)
169  hist(yyt)
170  ## Retomando os testes ##
171  ## Testes de Normalidade ##
172  shapiro.test(xxt)
173  shapiro.test(yyt)
174  ## Não houve normalidade das variáveis xx=tent_1_int e yy=tent_int_ok ##
175  ## Teste F:homogeneidade das variâncias ##
176  var.test(xxt,yyt)
177  ## Retoma-se os dados originais e executa o teste não paramétrico de Wilcoxon
    ↪ ##
178  wilcox.test(xx,yy)
179  #####
180  ## Há diferenças entre as médias ##
181  #####

```